Fall 2018

Investigating the Effect of Detecting and Mitigating a Ring Oscillator-Based Hardware Trojan

Lakshmi Ramakrishnan
Southern Methodist University, lramakrishnan@smu.edu

INVESTIGATING THE EFFECT OF DETECTING AND MITIGATING A RING OSCILLATOR-BASED HARDWARE TROJAN

Approved by:

Ping Gui
(Electrical Engineering)
Thesis Committee Chairperson

Jennifer Dworak
(Computer Science & Engineering)
Thesis Co-Advisor

Theodore Manikas
(Computer Science & Engineering)

INVESTIGATING THE EFFECT OF DETECTING AND MITIGATING A RING OSCILLATOR-BASED HARDWARE TROJAN

A Thesis Presented to the Graduate Faculty of the

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Master of Science in Electrical Engineering

by

Lakshmi Ramakrishnan

B.E., Electronics & Telecommunication Engineering, University of Mumbai

December 15, 2018

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my professor and advisor, Dr. Jennifer Dworak. Without her immense patience and deep knowledge, this thesis would not have been possible. It has been my utmost pleasure and deepest honor to have worked under such a amazing guide and mentor.

In addition to my advisor, I would also like to thank my committee members, Dr. Ping Gui and Dr. Theodore Manikas for their brilliant suggestions on how I can improve my work, as well as for being a huge part of my thesis journey. I would also like to thank Dr. Daniel Engels for his insightful comments and ideas on the research topic, as well as for helping me prepare for my defense.

I would like to thank my research friends Nada Alzaben and Yasamin Fozouni for always motivating me to work harder. I am also grateful for my wonderful research brothers and sisters Yi Sun, Saurabh Gupta, Fanchen Zhang, Hui Jiang, Nisharg Shah and Senwen Kan. I would also like to express my gratitude to my friends Wildon Menezes and Shuba Easwaran for their understanding and support. My appreciation also goes to my roommate, Disha Shah for her care and understanding for the entire period of us living together.

Finally, I would like to extend my deepest gratitude to my parents. It is because of their unending love and support that I was able to pursue and complete my master's degree.

Ramakrishnan, Lakshmi B.E., Electronics & Telecommunication Engineering, University of Mumbai, 2016

Investigating the Effect of Detecting and Mitigating a Ring Oscillator-Based Hardware Trojan

Advisor: Ping Gui

Co-Advisor: Jennifer Dworak

Master of Science in Electrical Engineering conferred December 15, 2018

Thesis completed December 15, 2018

The outsourcing of the manufacturing process of integrated circuits to fabrications plants all over the world has exposed these chips to several security threats, especially at the hardware level. There have been instances of malicious circuitry, such as backdoors, being added to circuits without the knowledge of the chip designers or vendors. Such threats could be immensely powerful and dangerous against confidentiality, among other vulnerabilities.

Defense mechanisms against such attacks have been probed and defense techniques have been developed. But with the passage of time, attack techniques have improved immensely as well. From directly observing the inputs or outputs, adversaries have tried, on multiple occasions, to extract data through other channels of data leakage, such as power, electromagnetic radiation and frequency, and have been very successful in doing so. This thesis investigates one such attack, known as hardware Trojan, where a ring oscillator is used as part of the Trojan to leak information regarding the logic value present at an internal circuit site through the frequency analysis of the power trace.

This thesis thus aims to expose a vulnerability of circuits, and also proposes a design technique to obfuscate the power trace to protect circuits against this kind of hardware threat. To test the efficacy of the Trojan, it is tested against power-related circuit protection mechanisms.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

This thesis is dedicated to my parents.

Chapter 1

Introduction

Integrated Circuits (ICs) are comparable to the brain when talking about most modern electronics systems. The creation of chips/ICs is a very large industry, and current demands for ICs require that they be mass-produced. Because chips are constantly increasing in complexity and size, the cost for designing and manufacturing chips is very high, and it would be very difficult for a single company to create them fully on their own [6]. Globalization, teamed with cost-effectiveness has led to various processes in the semiconductor industry, especially manufacturing, being outsourced to different countries all over the globe. Although economically viable, this level of outsourcing has led to the decentralization of control over the third-party facilities, thus exposing the ICs to several potential security threats.

The threat to hardware security is manifold. It could range from manipulating the circuitry to cause minor bit-flips in a mass-produced general-purpose IC, to leaking extremely sensitive data or causing military chips to malfunction during critical operations. For instance, researchers Sergei Skorobogatov and Christopher Woods detected a backdoor in the Actel/Microsemi ProASIC3 chips, which were at the time used as military-grade Field-Programmable Gate Array (FPGA). The backdoor would give the adversary access to the chip's configuration data and unencrypted bitstream, and would enable him to modify low-level silicon features or permanently damage the device [7].

Hardware security threats are usually classified as belonging to one of five categories. These include hardware trojans [3], side-channel attacks (SCA) [8], counterfeit chips [9], reverse-engineering, and IP piracy [10] [11]. If an adversary tries to obtain information on the processes taking place within a chip without destroying it, they would most likely inject a hardware Trojan horse or employ side-channel attacks. A Hardware Trojan Horse (HTH) is any threat to an IC's security through malicious modification of the original circuit [2]. Although it might be more likely that malicious modifications in ICs could be made at third-

party manufacturing facilities, it important to know that offshore semiconductor fabrication plants may not be the only threat to hardware security. An HTH may be inserted into an IC at one or more of the design stages, such as specification, Register-Transfer Level (RTL), layout, or fabrication [12]. This thesis investigates how a particular type of hardware Trojan horse may successfully leak information about a chip. In particular, we investigate a ring oscillator-based Trojan designed to leak information regarding the logic value at a circuit site through the circuit's power trace.

Even when Trojans are not employed, it is possible to deduce information about circuit operation by measuring and recording the power trace. Because encryption circuits are particularly vulnerable to such attacks, several countermeasures have been employed to protect them. One of the major SCA threats to encryption circuits is through the Differential Power Analysis (DPA) attack, and there have been many publications that discuss how to protect circuits from such attacks. These countermeasures are of two main categories. The first kind tries to make the power consumption equal throughout the power trace, thus making the power consumption independent of intermediate operations and data values. The second kind tries to obscure the power consumption by introducing extra gates for additional switching activity, thus providing the attacker with false information on the power consumption.

Although DPA-mitigation techniques work quite well, none of them are perfect, as the resulting power trace generally is not completely independent of intermediate data values or operations in the circuit.

This thesis is an attempt to analyze the vulnerability of circuits when more than one kind of attack on hardware security may be implemented. This is achieved by implementing a DPA-mitigation technique, after which the circuit is then subject to a power-based hardware Trojan attack. We will show that the proposed Trojan can leak information even in the presence of a selected DPA countermeasure. Thus, this thesis also proposes a protection mechanism against this kind of Trojan attack.

1.1 Contribution

For preventing the leakage of side-channel information through the power trace for a ring oscillator-based HTH, the frequency of switching needs to be obscured or indecipherable. In this thesis, the vulnerability of circuits to frequency analysis of power side-channel attacks is highlighted by the implementation of the HTH in benchmark circuits. The effects of the ring oscillator's chain length are also investigated.

To counter the frequency analysis attack, a method of obscuring the frequency is suggested. By the usage of a technique proposed in this thesis, information on logic values at an internal circuit point will be obscured even in the presence of frequency analysis of the power trace. The functionality and efficacy of this technique are verified by implementing it in the benchmark circuits, and by performing frequency analysis.

1.2 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 provides background on hardware Trojan horses, and discusses the techniques used for Trojan detection. Chapter 3 explains in detail how differential power analysis attacks work and also provides an overview of the mitigation techniques. Chapter 4 talks about the experimentation setup, as well as the results obtained. This chapter also proposes a novel technique for obscuring the power trace as a countermeasure for an information-leaking ring oscillator-based hardware Trojan. Chapter 5 describes the conclusions drawn from the experimentation, and provides directions for future work.

Chapter 2

Background on Hardware Trojans

This chapter introduces the concept of hardware Trojan horses and investigates how HTHs can be a major threat to integrated circuits. Some techniques for HTH detection are also discussed, and the design of the HTH used in this thesis is shown.

2.1 Hardware Trojan Horse

The chip design and manufacturing process, as shown in Fig 2.1, is fragmented and is spread out across the globe [8]. Because it is so spread out, the entire process is decentralized, with the vendors having little actual control. In recent years, there have been several accounts of incidents involving adversaries attempting to attack either the functionality of a chip, or create a backdoor to it. The Syrian nuclear bombing is one such example. In 2007, state-of-the-art Syrian radars failed to detect an incoming Israeli airstrike. Further probe concluded that the reason for the failure in their detection system was the result of a backdoor that was created in one of the system's chips [13] [14]. Such kinds of invasions at the hardware level are known as hardware Trojan attacks, and chips may be susceptible to hardware Trojan insertion at one or more levels of the design and manufacturing flow shown in Fig 2.1.

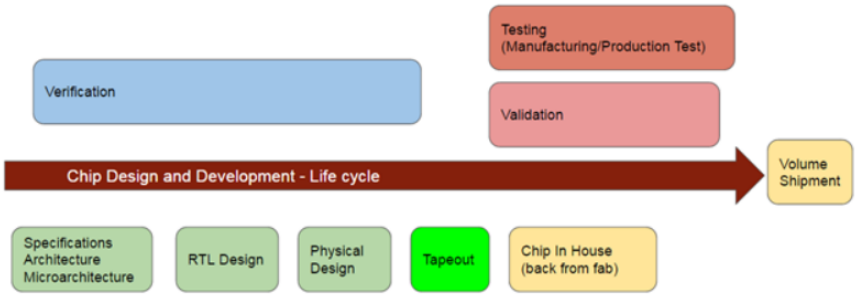

Figure 2.1: Chip design and development [1]

A Hardware Trojan Horse (HTH) is any malicious addition to or modification of a circuitry or hardware system. As mentioned previously, this malicious addition or modification could be of any kind, ranging from introducing a minor fault in the circuit, to causing the failure of critical operations in the circuit. Because hardware Trojans can be of many kinds, and there is no single solution for detecting or preventing against all the different types of HTHs, a metric of classification had to be developed for the application of appropriate protection or detection mechanisms. In 2010, Wang et al. introduced a detailed taxonomy for the classification of HTHs based on their physical, activation, and action characteristics, and this taxonomy is as shown in Fig 2.2 [2] [15].

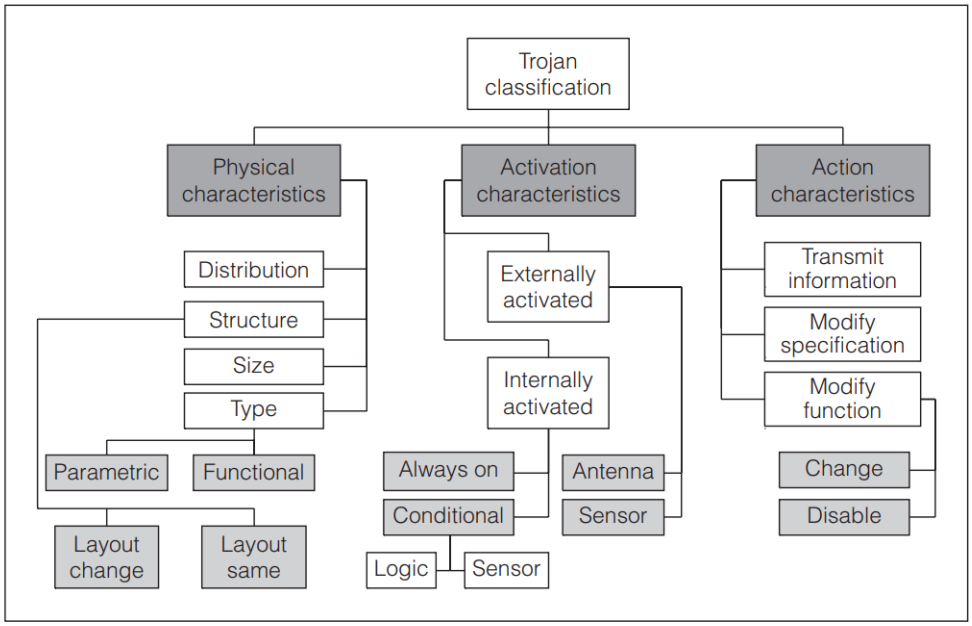

Figure 2.2: Hardware Trojan taxonomy [2]

In Fig 2.2, "physical characteristics" refers to how the physical characteristics of the HTH may be manifested at the hardware level. Activation characteristics refers to the criteria under which the HTH may be activated to carry out its malicious functionality, and action

characteristics refers to the malicious functionality of the HTH. Although the categories are clearly defined, an HTH may be classified under more than one category.

2.2 Hardware Trojan Implementation

The HTH implemented in this thesis is designed with consideration for the HTH structure. The general structure of an HTH is as outlined in Fig 2.3. Most HTHs consists of a trigger and a driver/payload [3]. It must be noted that although a lot of hardware Trojans may follow the general structure, not all Trojans may be structured so. In fact, some HTHs may additionally contain storage, as classified by Alkabani and Koushanfar [16], and the payload need not always be connected with a circuit signal S.

The trigger is made up of the circuitry that would activate/deactivate the HTH under specific criteria, depending on its mode of operation. The driver/payload is made of the circuitry that would perform the malicious operation of the HTH (bit-flip, leakage of information, etc.).

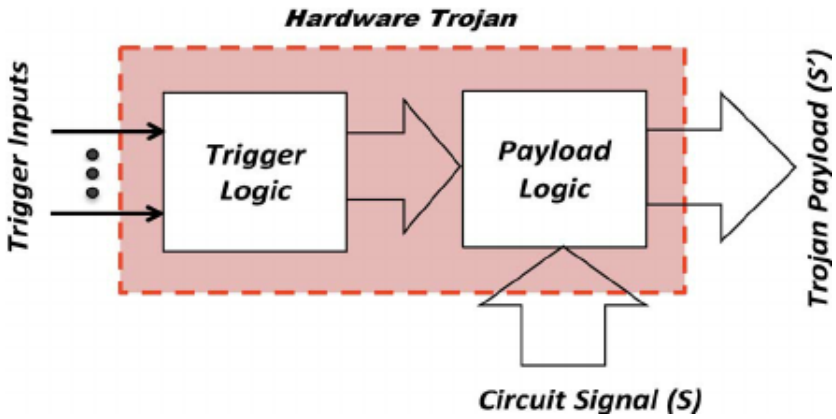

Figure 2.3: General structure of a hardware Trojan horse [3]

For designing the payload for the HTH, the effects of the HTH need to be taken into consideration. The HTH must not be observable in the power trace as changes in the power trace may be observed, especially when compared with the power trace of a "golden" circuit.

If the effects of the HTH are visible in the power trace, that could cause the circuit to fail during manufacturing tests. To make sure that the HTH goes undetected, the Trojan trigger should be such that it causes switching only under the conditions that are unlikely to be seen during test, and the Trojan payload must be such that there is no switching when the trigger is not activated. It is well-known that ring oscillators may be used as part of a HTH in a chip. Traditionally, ring oscillator-based HTH have been used to destroy infected chips by raising their core temperature [17].

In this thesis, we investigate a particular type of hardware Trojan that includes a ring oscillator (RO) in the payload. The ring oscillator is used here to leak information from the chip rather than destroy it.

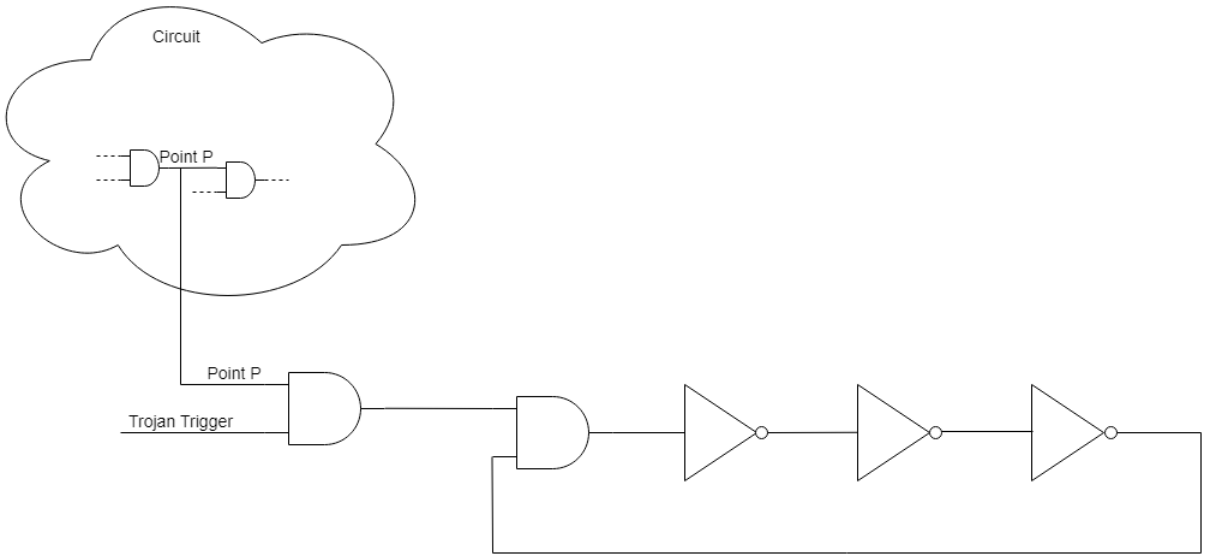

Figure 2.4: Idea for the HTH

The design for the HTH used in this thesis is shown in Fig 2.4. The HTH leaks information about a point P by enabling its logic state to be observed more clearly in the power trace. When the Trojan trigger is enabled and the point P is at a logic high value, the ring oscillator is enabled. Because the ring oscillator oscillates at a high frequency, usually higher than the

frequency of the clock, the behavior of the ring oscillator can be observed clearly with the appropriate tools. The oscillations of the ring oscillator would indicate that the point P being observed, as well as the trojan trigger, are at a logic high. This kind of HTH can hence be used for easy extraction of data, such as values that could help extract the secret key otherwise spy on the chip.

2.3 Hardware Trojan Detection

Over the years, several techniques have been implemented for protection against of various kinds of HTHs. These techniques, as shown in Fig 2.5, were briefly classified into three main categories by Xiao et al. as "Trojan Detection", "Design for Trust" and "Split Manufacturing for Trust" [18].

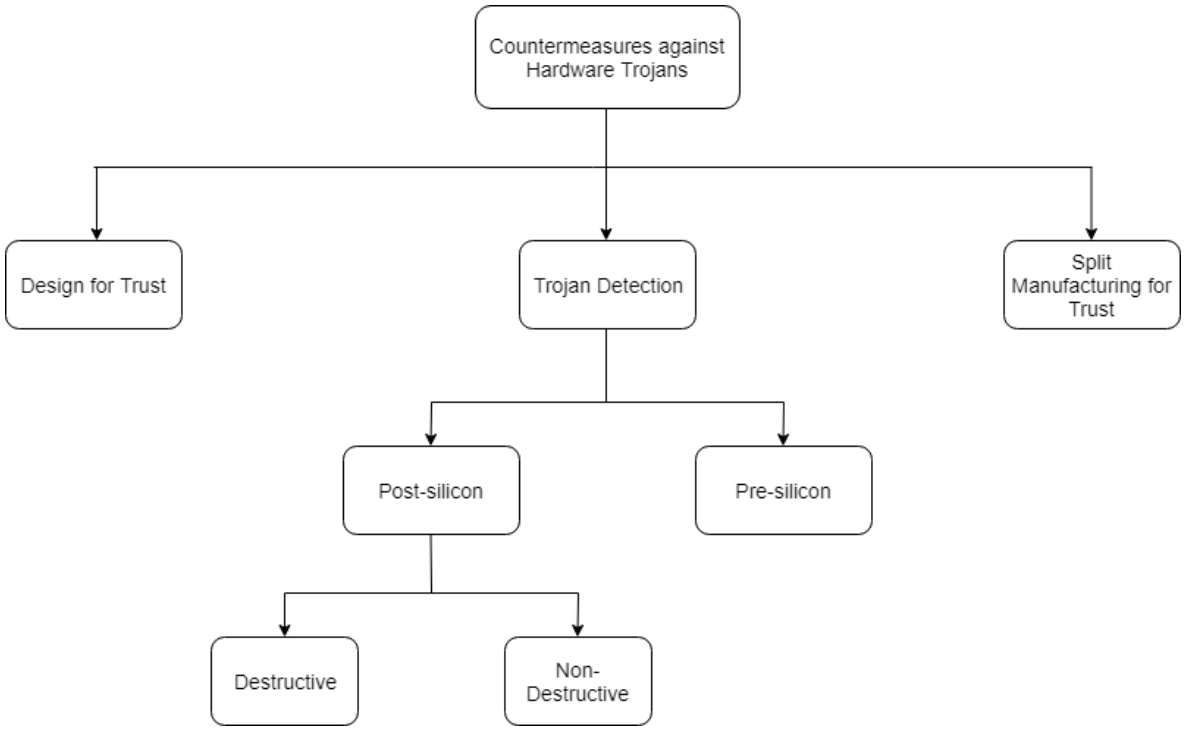

Figure 2.5: HTH countermeasures

2.3.1 Trojan Detection

Trojan detection refers to direct methods used for HTH detection without supplementary circuitry. It is either performed at the pre-silicon (design) level or at the post-silicon (post fabrication) level.

2.3.1.1 Pre-Silicon Trojan Detection

To detect the presence of HTHs, several researchers have employed functional validation as a detection technique by employing random and constrained-random test vectors. Functional validation is the technique used to confirm the functional correctness of the circuit by checking the logic value at the output pins [19] [20]. Other techniques look at toggle coverage to identify suspicious nodes, and in come cases try formal verification [21] [**?**].

2.3.1.2 Post-Silicon Trojan Detection

Post-silicon Trojan detection is classified as either destructive or non-destructive. For post-silicon Trojan detection, destructive techniques are not generally used because of the following two reasons:

1. Although malicious modifications in the ICs may be more detectable using this method, it would normally take several weeks, to a couple of months for this process.

2. In the end, the IC will be rendered unusable and because the malicious modification may not exist in all of the fabricated ICs, the information obtained from this destructive approach will only be relevant as a single sample.

As for the non-destructive approaches, functional test refers to using test patterns to detect faults in a circuit. Although functional tests are not directly used to detect HTHs, some of the patterns generated for functional test may trigger the Trojan, and hence activate its mechanism during the test phase. However, an attacker will generally design a Trojan that will be unlikely to be caught using functional test. Structural tests, such as those generated with Automatic Test Pattern Generator (ATPG) tools may also detect Trojans, but once again, an attacker can try to design a Trojan to avoid likely detection with this approach.

The other non-destructive approach, side-channel signal analysis, consists of several different types of Trojan detection approaches. Side-channel analysis has been proven to be especially effective at detecting HTHs by analysis of delay [22], transient power [23], and current [24]. Side channel analysis is generally more effective than logic test-based approaches because even inactive Trojans may have a side channel effect.

2.3.2 Design for Trust

Design for trust (DFTr) refers to the techniques that can be implemented to protect circuits against the insertion of hardware Trojans and facilitate Trojan detection at the same time. Such techniques may rely on making modifications to the IC design flow. DFTr usually comprises three techniques: logic encryption/obfuscation for hiding functionality, IC camouflaging to prevent reverse-engineering through layout, and hardware Trojan activation, usually through test patterns [25] [26].

2.3.3 Split Manufacturing for Trust

Split manufacturing for Trust was a technique suggested by leading fabless semiconductor companies, and it refers to the splitting of the manufacturing process into sub-processes, which may be performed at different trusted or untrusted facilities. In Split Manufacturing for Trust, the layout is split into two layers, namely Front End Of Line (FEOL) and Back End Of Line (BEOL) based on which metal layers are used for implementation. The FEOL layer consists of transistors and the lower metal layers and may be fabricated at an untrusted foundry. The BEOL layer consists of the higher metal layers and are fabricated at a trusted foundry. After fabrication, the two wafers are then aligned and integrated together. Because the untrusted foundry will have no information on the BEOL layer, the process of split manufacturing is considered to be very secure [27].

Chapter 3

Power Analysis Attack and Mitigation

This chapter elaborates upon a (encryption) circuit's protection mechanisms by introducing a well-known threat, known as a Side Channel Attack (SCA) [4]. The Trojan design for this thesis will then be tested against the circuit's protection mechanism.

When talking about electronic circuits, an unsophisticated attacker may assume that data can only be leaked through the output pins of a chip as this is what makes up the most basic and direct type of attack. But with the passage of time, adversaries have come up with several new attack techniques. One such technique is called a Side Channel Attack (SCA). Apart from the output pins, a circuit may also leak information through other physical characteristics. In hardware security, a side-channel attack is an attack that takes advantage of information gained through analysis of the physical measurements that are related to a circuit's operation and implementation characteristics [8]. There are several different types of side channel attacks, as outlined in Fig 3.1 [4]. They are broadly classified under there following categories:

1. Power monitoring attack
2. Timing attack
3. Electromagnetic attack
4. Differential fault analysis attack
5. Acoustic cryptanalysis attack.

Amongst all of these side-channel attacks, the power monitoring attack will be discussed in detail for the purpose of this thesis as it is one of the most common attacks. A power monitoring attack is a type of side-channel attack, wherein the attackers extract crucial

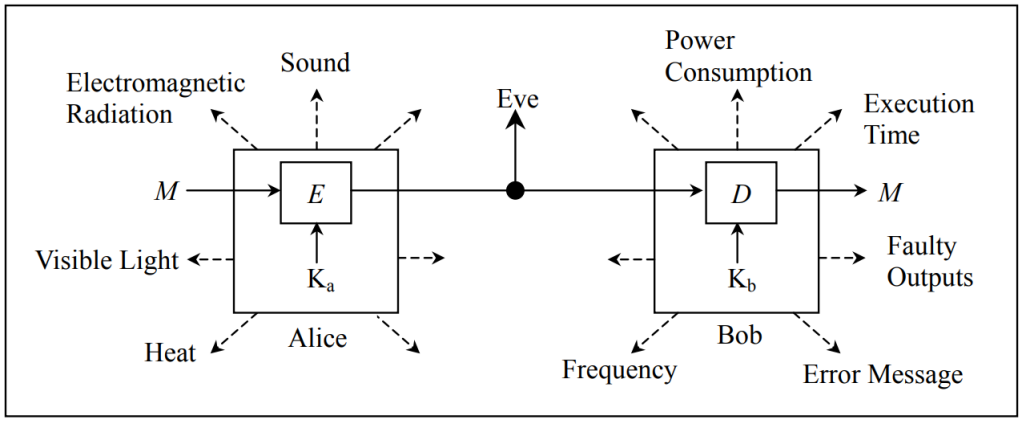

Figure 3.1: Cryptographic model including side-channel effects [4]

information about a circuit's operation based on observation of the instantaneous power consumption. This is made possible by the direct correlation between power consumption and switching activity of logic gates and transistors within a digital circuit. To practically measure a circuit's power consumption, a small resistor is inserted in series with the power source. The current is obtained by calculating the potential difference between the two nodes of the resistor. The power model for CMOS gates is given by Kocher et al [5] as:

$$P_{total} = P_{switching} + P_{short-circuit} + P_{leakage}$$

Here, the total power is the sum of dynamic power and leakage power, and the dynamic power is the sum of switching power (P_{switch}) and short-circuit power ($P_{short-circuit}$). $P_{switching}$ relates to the total switching activity in the circuit, i.e. transistor switching that would cause the capacitor to get charged or discharged. The formula for $P_{switching}$ is given as [59]:

$$P_{switching} = \alpha.f.C_{eff}.Vdd^2$$

Here, α = switching activity, f = switching frequency, C_{eff} = effective capacitance, and Vdd = supply voltage. Here, $\alpha = 1$ if the signal is a clock, and α is 1/2 if the signal switches once per cycle.

$P_{short-circuit}$ refers to power consumption during an instantaneous short-circuit connection

between the supply voltage and the ground at the time the gate switches state due to a brief moment of simultaneous PMOS and NMOS switching. The equation for $P_{short-circuit}$ is given as:

$$P_{short-circuit} = I_{sc}.Vdd.f$$

where I_{sc} = short-circuit current while switching, Vdd = supply voltage and f = switching frequency.

$P_{leakage}$ is the static power consumption of the circuit. Static power is the baseline power that may be consumed by the components of the circuit when there is no circuit activity. The equation for $P_{leakage}$ is given as

$$P_{leakage} = (I_{sub} + I_{gate} + I_{junct} + I_{contention})Vdd$$

where I_{sub} = subthreshold leakage current, I_{gate} = gate leakage current, I_{junct} = junction leakage current, $I_{contention}$ = contention current, and Vdd = supply voltage.

When there is no switching activity in the circuit, only leakage power will be consumed. When there is switching activity, dynamic power will also be consumed. As dynamic power is driven by the switching activity (i.e. total # of gates switching), the total dynamic power consumption during this period will be proportional to the switching of the gates. Hence, coupled with basic knowledge of the circuit, analyzing the power trace could give the adversary a significant amount of information regarding what kind of operation is taking place within the circuit.

There are two main kinds of power analysis attacks, both of which share the same principle, but differ in terms of purpose, complexity and method of execution. They are:
1) Simple power analysis (SPA) attacks
2) Differential power analysis (DPA) attacks.

Simple power analysis attacks were a commonly-used attack mechanism before the introduction of DPA. The differential power analysis attack is the more commonly-used attack method now, and employs SPA as the first step. The two attacks are described in detail in the following sections. Because these attacks are commonly applied to encryption circuits, the discussion will focus on how these attacks may be used to extract an encryption key.

3.1 Simple Power Analysis Attack

Simple Power Analysis (SPA) attacks involve direct interpretation of the power trace to reveal information about device operation during a single encryption/decryption operation. As already note, this information can be used to deduce the secret key. Fig 3.2 shows the current trace for a single 16-round encryption operation for a DES circuit from a typical smart card. The 16 rounds of encryption are clearly visible by observing the 16 distinct peaks in the trace [5].

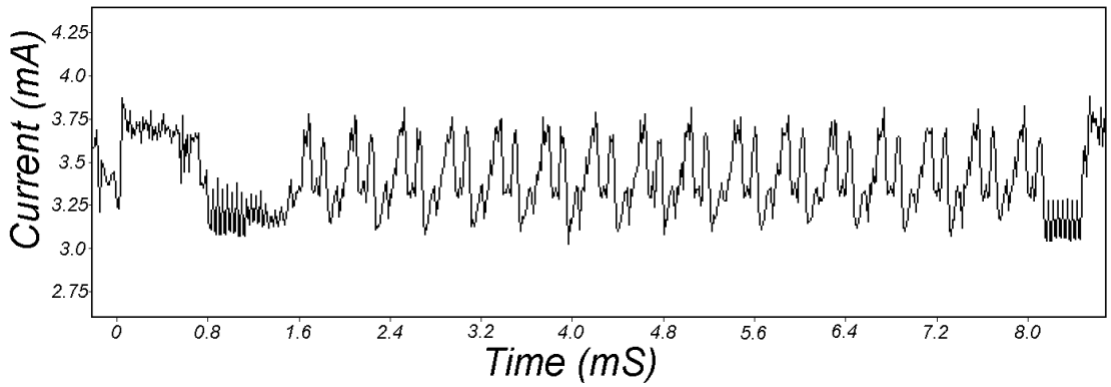

Figure 3.2: Simple Power Analysis trace showing DES encryption rounds [5]

With SPA, it becomes possible for the attacker to extract a lot of information if they have a high power oscilloscope and can zoom in to the power trace where encryption is taking place. For example, zooming into encryption rounds 2 & 3 of Fig 3.2 results in Fig 3.3. Hence, more features are now visible by observing the power trace. For example, the arrows in Fig 3.3 indicate rotation operations occurring once in round 2, and twice in round 3. Additionally, even more information about operations can be gained by observing the power trace in greater resolution [5].

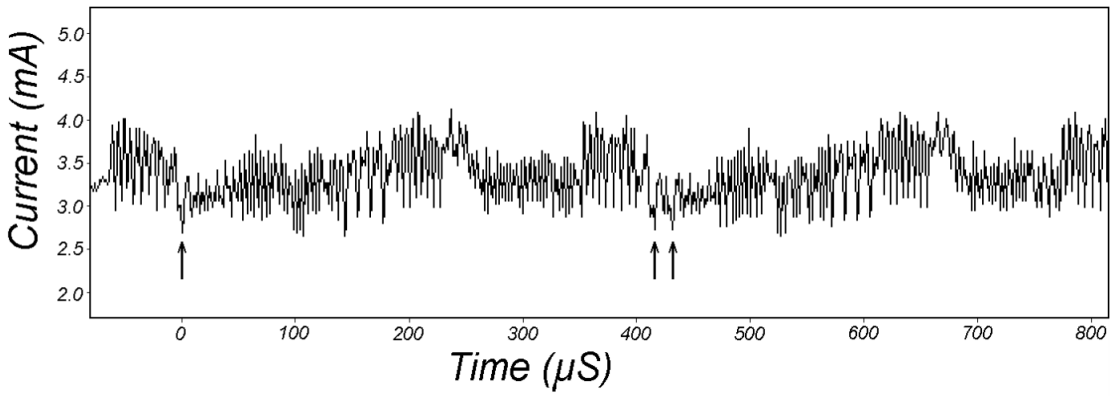

Figure 3.3: Simple Power Analysis trace showing DES rounds 2 & 3 [5]

3.2 Differential Power Analysis Attack

With SPA, smaller values for variation in the power trace might mean the variations get overshadowed by measurement errors and noise. To overcome this problem, Differential Power Analysis (DPA) attacks are implemented by attackers. In DPA, statistical techniques tailored to the target algorithm would be used over a large number of power traces to obtain the correlation between instantaneous power consumption and the operations taking place within the circuit. The benefit of implementing DPA is that it is much more sophisticated than SPA, and hence more difficult to prevent.

According to S. Mangard et al., all DPA attacks, no matter how complex, follow a general strategy. It consists of five steps, which are as follows [28] [29]:

1. Multiple power traces from encryption/decryption runs are gathered and related to the corresponding value of d, where d is a known non-constant data value, usually the plaintext. The result is stored as a vector $d_i = (d_{i,1}, d_{i,2}, ..., d_{i,D})'$, where D denotes the total number of data blocks that are encrypted/decrypted and d_i denotes the data value in the i^{th} encryption/decryption run. The attacker also records the time period corresponding to each run and this result is stored as a vector $t_i^{'} = t_{i,1}, t_{i,2}, ..., t_{i,T}$. Here, T denotes the length of the trace, and t_i is the time value in the i^{th} encryption/decryption run. The traces can then be stored as a matrix T of size D x T. Because in DPA, the difference function of the power traces is taken into consideration, it is crucial that

the power traces for each of the runs are aligned correctly. This can either be done using a trigger signal to enable the capture of the power traces by the oscilloscope, or by using graphical techniques for alignment.

2. **T**he intermediate result should be represented as a function $f(d, k)$ where d is a known non-constant data value, usually the known plaintext or generated ciphertext, and k is a small part of the key.

3. **H**ypothetical intermediate values for every possible choice of k is calculated and represented as a vector of the form $K = (k_1, k_2, ..., k_N)$, where N denotes the total number of possible choices for k. The elements of vector K are called key hypotheses. Hence, with the key hypotheses K and known vector d, the adversary is able to calculate the hypothetical intermediate values for all of the encryption runs and key hypotheses. This calculation is used to create a matrix V of size D x N. Because the column vectors for matrix V correspond to the key hypotheses, the goal of DPA is to find out which key hypothesis has been processed during the D runs. The corresponding key hypothesis would then be the key.

4. **T**he power consumption of the device is simulated for each hypothetical intermediate value to obtain the hypothetical power consumption value. Here, matrix H is used to denote instantaneous power consumption values. By mapping matrix V to H, the attacker compares the hypothetical intermediate values to the hypothetical power consumption values of each key value with the power trace at each position. The power models usually employed to map V to H are the Hamming-distance and the Hamming-weight models.

5. **A** matrix R of size N x T is now created. Here, R is the comparison of each column h_i of matrix H with each column t_j of matrix T. This means that the adversary compares the hypothetical power values with respect to each key value, with the recorded trace. The comparison is usually performed by estimating the correlation coefficient. For higher values of $r_{i,j}$, the values of matrix H and T map better. This means that, when the value of the key is correct, a high correlation between the value of matrix T and

corresponding value of matrix H will be obtained.

In DPA, because power consumption values obtained through many iterations of power analysis are averaged out, the effect of noise and measurement errors is reduced.

3.3 Countermeasures Against Power Analysis Attacks

Ever since Kocher et al. [5] proposed SPA and DPA attacks, several different types of countermeasures have been developed. Although different in nature and functionality, they usually follow one of the two main types of mitigation techniques [29]. These are listed as:

1. **Hiding:** This technique focuses on making the instantaneous power consumption independent of internal computations.

2. **Masking:** This technique focuses on obscuring the power trace in such a way that the instantaneous power consumption would not be directly reflective of the circuit's internal operations.

3.3.0.1 Masking

The masking countermeasure aims to make power consumption independent of the intermediate values by randomizing the intermediate values being processed at the algorithmic level. In masking, each intermediate value is concealed by an internally-generated random value (mask) such that:

$$v_m = v * m$$

Here, the $*$ operation usually depends on the operations involved in the cryptographic algorithm. It usually represents EX-OR function (Boolean masking), modular addition (arithmetic masking) or modular multiplication (arithmetic masking). The masks are usually directly applied to the plaintext or key bits. Because the attacker does not have any information on the value of the mask m, it becomes difficult for the attacker to relate the power consumption to what kind of operation is being performed.

3.3.0.2 Hiding

The hiding countermeasure aims to sever the link between intermediate values and power consumption. There are two main approaches to this countermeasure:

1. Randomize power consumption for each clock cycle

2. Equalize power consumption for all operations for each clock cycle

Although perfectly equalized power consumption or perfectly randomized power consumption cannot be achieved in reality, the hiding techniques commonly used are quite effective.

To randomize power consumption, three main techniques are used, and they are listed as follows:

1. Random insertion of dummy operations (Time Dimension)

2. Shuffling (Time Dimension)

3. Increasing noise (Amplitude Dimension)

Equalization of power has only one category, and that would be to reduce the signal level.

At the cell-level, Dual-Rail Precharge (DRP) logic styles are used to counter DPA attacks by making the power consumption of the cells independent of the processed data and the performed operations. DRP logic styles focus on making power consumption equal throughout all operations for each clock cycle by using Dual-Rail (DR) and precharge logic.

There are two main DRP logic styles. The first logic style is called Sense Amplifier Based Logic (SABL), and the second logic style is called Wave Dynamic Differential Logic (WDDL), which is a refinement of a technique called Separated Dynamic Differential Logic (SDDL).

In 2004, Tiri et al. [30] published the first design methodology to use Differential Dynamic Logic (DDL). The goal of DDL is to make power consumption independent of switching activity. This is done by making either the switching activity constant, or the load capacitance constant.

For this thesis, SDDL is used as the DPA-mitigation technique. The basic logic behind SDDL is highlighted in Fig 3.4. Here, "n_prch" represents the $\overline{Precharge}$ signal, "n_A" represents $\bar{A}$, and "n_B" represents $\bar{B}$.

The idea behind SDDL is that for every AND gate, there is a complimentary OR gate that also provides switching when the AND gate does not. This leads to having at least one

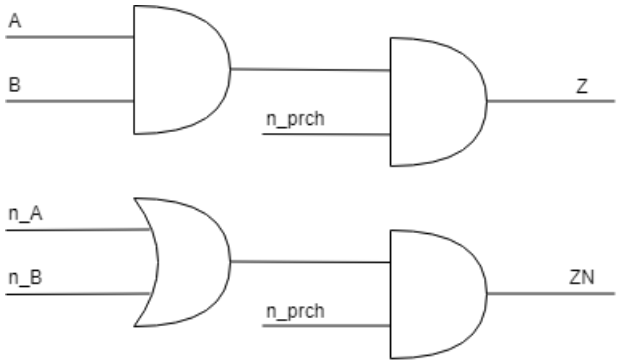

Figure 3.4: SDDL implementation of AND gate

switching activity per clock cycle. In case the inputs to the gates are repeated, there will usually be no switching activity, which leads to the introduction of the precharge signal. The precharge signal is connect to the second two AND gates. This means that if the precharge signal is deasserted, i.e. in the precharge phase, the outputs of the precharge-controlled AND gates will be forced to become logic 0, regardless of their previous state. When the precharge signal is at logic 0, i.e. in the evaluation phase, the AND gates connected to $\overline{Precharge}$ will be able to propagate the value at their other input terminal. This means that even if there is no change in inputs for consecutive clock cycles, there will be some switching activity due to the effect of the $\overline{Precharge}$ signal.

For implementing SDDL in this thesis, the circuits C432 [31] and C499 [32] were synthesized with only the basic AND, OR and INV gates, after which cell substitution was performed in order to include the complimentary logic. This was performed using a Perl script where for each AND/OR gate, the complimentary SDDL gates were added directly to the synthesized Verilog with the appropriate wire connections.

Chapter 4

Results

Chapters 2 and 3 provided background on the creation of hardware Trojans, countermeasures against HTHs, and power analysis mitigation techniques. This chapter focuses on the concurrent implementation of these techniques. For this purpose, the ISCAS HLM benchmark circuits, C432 [31] and C499 [32], have been employed. For the DPA-mitigation technique in our experiments, we are using SDDL for most of the analyses.

In out experiments, the hardware Trojan has been designed to be able to observe the current logic value at a point P in the circuit. A DPA-mitigation technique (SDDL) is applied to the circuit and the efficacy of the HTH in leaking information is measured when SDDL has been added. The goal is to determine whether DPA mitigation has an effect on the detectability or efficacy of the HTH. Additionally, the proposed method for obfuscating the frequency of switching in the power trace is also analyzed by implementing it against the HTH.

4.1 Methodology

The simulation experiments included in this thesis followed the protocol below:

1. As part of the general setup, all the logic gates in the NanGate 45nm library [33] were imported into Cadence Virtuoso [34].

2. Obtained Verilog code for the circuits from the ISCAS HLM website [35].

3. Modified the NanGate library to include only the basic 2-input AND, 2-input OR and INV gates and created a database file for the new restricted library using Synopsys Library Compiler [36] (A tutorial for this can be found at [37]).

4. Synthesized the Verilog for each circuit using modified NanGate 45nm library with Synopsys Design Compiler [38] (A tutorial for this can be found at [39]).

5. If a HTH or the proposed frequency obfuscation circuitry needed to be added to the circuit, it was done post-synthesis and added directly to the synthesized Verilog. Additionally, if SDDL needed to be added to the circuit, it was also done post-synthesis through means of cell substitution using a Perl script.

6. A stimulus file was then created using Perl to provide the simulator with instantaneous values for the logic inputs.

7. Timing analysis using Synopsys PrimeTime [40] was performed to ascertain that the slack was non-negative. If the slack was negative, the time period between two subsequent input signals was increased and tested again until positive slack was obtained.

8. The synthesized Verilog file was then imported into Cadence Virtuoso and set up in the ADE L environment [58] to automatically generate a SPICE netlist from the synthesized Verilog schematic. The SPICE netlist will later be used for HSPICE simulation.

9. After the generation of the SPICE netlist, a Perl script was used to automatically generate .MEASURE statements to calculate the average power every 10ps. These .MEASURE statements were then added to the SPICE netlist.

10. SPICE simulation was then performed using Synopsys HSPICE [57]. This simulation generated the files .mt0 and .tr0 (among others).

11. The .tr0 file contained information on the instantaneous logic values of all the nets in the circuit. When opened with Synopsys WaveView [56], we could use the .tr0 file to observe the instantaneous logic value of point P.

12. The .mt0 file contained the results of the .MEASURE statements. A Perl script was written to parse the data in .mt0 file into .csv format. The graph of power vs time was then plotted using LibreOffice [41].

13. For frequency analysis of the power trace, a Perl script was written to parse the x (time) and y (power) values to .txt format.

14. A .m file was written to import the .txt files into MathWorks MATLAB [42] and filter the signal using a high-pass filter.

4.2 Hardware Trojan Trigger

The general structure of a Hardware Trojan Horse (HTH) was introduced in Ch 2. The trigger circuit designed for this thesis shown in Fig 4.1. The payload of the HTH can be triggered and untriggered based on the input patterns applied to the trigger circuit. Considering an initial state of logic 0 for Q, if the trigger pattern is applied once, i.e. if all the inputs are set to a logic 1, the D flip-flop (DFF) will toggle, causing Q to now hold the logic value 1. If the trigger pattern is applied again on another clock cycle, the DFF will toggle again, causing the value of Q to become logic 0. This toggling action will repeat itself depending on how frequently the trigger pattern is applied. If the inputs to the Trojan trigger are at a logic 1 for multiple clock cycles, the DFF will keep toggling. When Q is holding the value of logic 1, the HTH payload is activated for the time that Q holds its value. When Q toggles to a logic 0, the HTH payload is deactivated until Q toggles again.

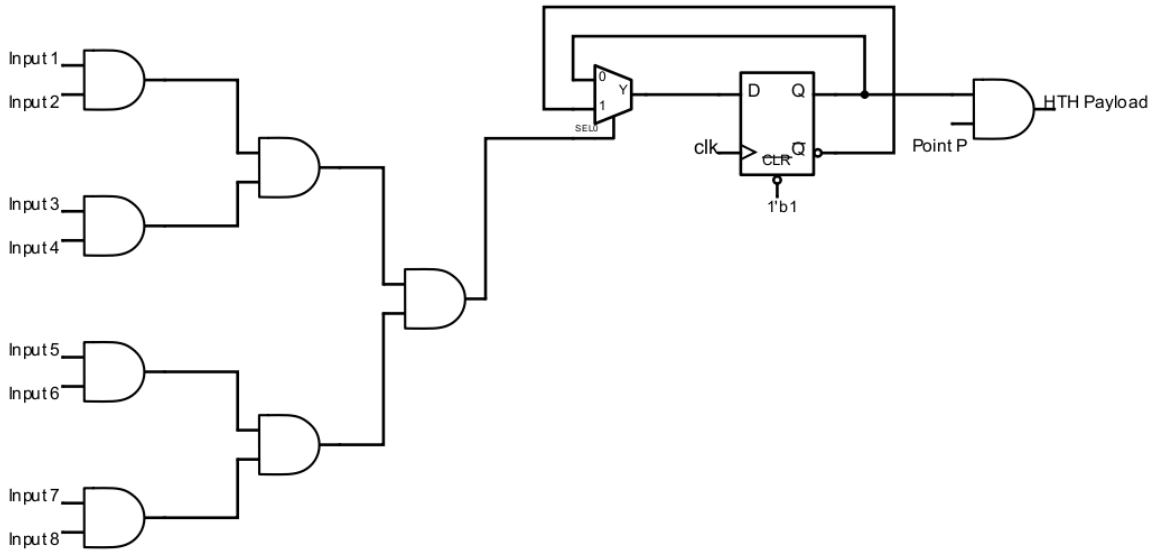

Figure 4.1: Basic concept for trigger circuit

The benefit of using this kind of trigger is that the triggering of the HTH can be controlled

externally by the adversary.

The input signals shown in Fig 4.1 can be chosen by the adversary to be unlikely to be activated during functional or structural test, and they could either be taken from Primary Inputs (PIs) or from intermediate nodes. The benefit of using PIs to activate the HTH is that the adversary will have easy control over when the Trojan gets triggered and untriggered, but this would also mean that there would be a higher chance of the HTH being activated during test because PIs are perfectly controllable, when compared with nodes that are less controllable. If the logic value from intermediate nodes are used as the input signals for the Trojan trigger, the triggering of the HTH may be comparatively more difficult for the adversary to control because additional conditions may need to be satisfied at the primary inputs to justify the logic values needed for the trigger. The benefit of using intermediate nodes though, is that it would be relatively more difficult to trigger the HTH during test, but this will depend on the controllability and the 1's probability of the intermediate nodes [43].

To simulate the Trojan trigger to measure its maximum and minimum power consumption, we implemented the HTH trigger using the NanGate 45nm PDK library. The total power consumption of the trigger circuit will vary with respect to the switching activity within the trigger. The power consumption of the Trojan trigger with different amounts of switching activity is as shown in Fig 4.2. Here, the minimum power consumption is about 0.5mW. At about 118ns, for when all of the gates are switching, there will be maximum switching activity and the power consumption will be approximately 2mW. This finding implies that even if a single AND gate in the Trojan trigger is switching, there will be an increase in the overall power consumption for that particular time period by about 0.5mW.

The determining factor for how easy or difficult it may be for a test engineer to detect this HTH by using power-based techniques would depend on the size of the original circuit and the number of HTHs implanted. This is because if the adversary wishes to leak information on multiple sites in the circuit, the adversary would require a modified trigger circuit for each implementation of the HTH, which would thus cause additional switching activity and hence cause additional power consumption. The percentage of total circuit's power that the HTH would consume would depend on the size of the original circuit, thus making the HTH less detectable in a larger circuit. Because DPA-mitigation techniques usually increase the

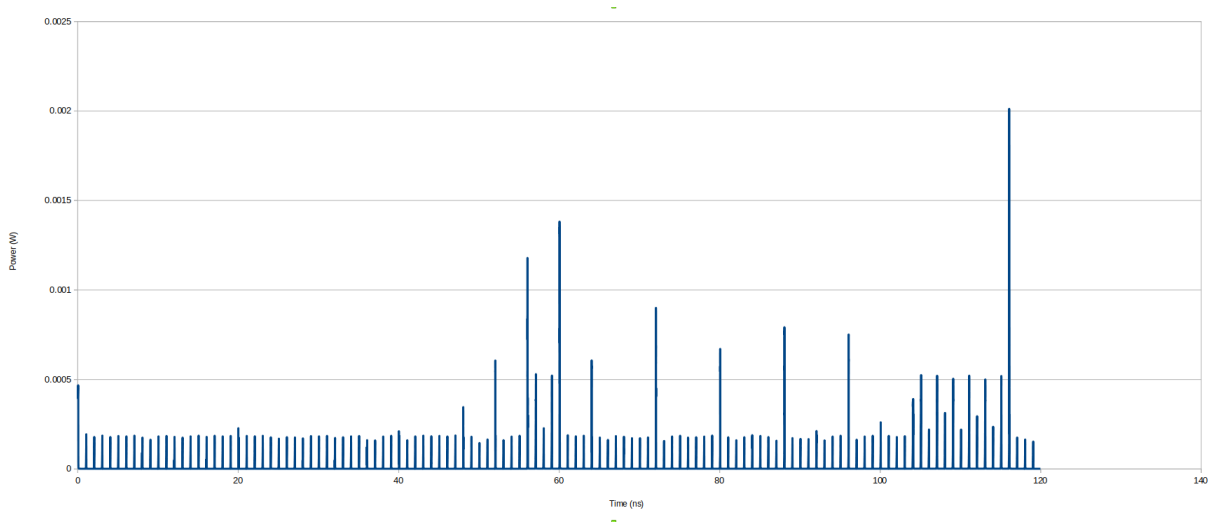

Figure 4.2: Power trace of HTH trigger for multiple input combinations

number of cells in a circuit, the Trojan trigger would consume a lower percentage of total power consumption in a circuit with DPA protection, hence making the detection of the HTH comparatively more difficult through power analysis in such a circuit.

Additionally, the probability of the HTH being set active during test will depend on multiple factors. They could include the number of inputs the Trojan trigger contains, the location at which the HTH is inserted, and the controllability of its inputs. For the trigger circuit shown in Fig 4.1, the probability of the Trojan being triggered in isolation, considering a 1's probability of 0.5, is: $P_r = (0.5)^8$. Therefore, $P_r = 0.00390625$, or 0.39%. Additionally, the probability of triggering the HTH will be lower for a larger trigger pattern.

For the purpose of triggering the payload of the HTH, an always-on trigger was used for all experimentation purposes. Here, the HTH is always enabled by adding a connection for the power line in place of the trigger circuit. The payload in Fig 4.1 is directly connected to the AND gate with the inputs VDD and Point P in this case. Doing this instead of creating a trigger for the HTH will make the HTH vulnerable to detection as:

1. The HTH will leak information whenever the circuit is powered on. This means that the malicious effects of the HTH may be easily detected through power and/or frequency analysis in the parametric test stages as the payload will be automatically enabled [44].

2. If the adversary wishes to use the HTH to observe multiple points at multiple sites in the circuit, it may not be possible for them to do so. If two or more RO-based HTHs with "always-on" trigger are introduced to the circuit, the adversary will not be able to view the effects of each of the multiple ROs in isolation. This is because although the effect of the ROs may be visible through frequency analysis of the power trace, it will be impossible to tell which RO was oscillating.

Because we have already analyzed the impact of the Trojan trigger on the power trace in isolation, the rest of the thesis will be using an always-on trigger so that we can directly analyze the effect of the payload. Another reason for using an always-on trigger for experimentation purposes is that it is possible for an adversary to use a different kind of trigger, but we are primarily interested in the behavior of the ring oscillator-based payload.

4.3 Trojan Payload

The payload for the HTH was introduced in Chapter 2. In this section, the attributes of the payload, such as frequency and power consumption, will be analyzed.

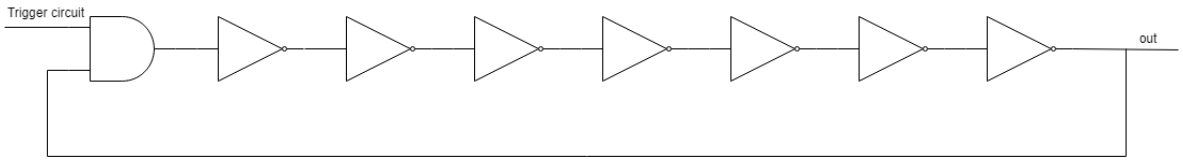

Figure 4.3: Ring oscillator as HTH payload

A ring oscillator, as shown in Fig 4.3 is a combination of an odd number of inverting gates that leads to constant switching activity in the circuit. For this purpose, a ring oscillator would always be made up of an odd number of INVs or NANDs connected that form a ring. Because ring oscillators provide high-frequency switching when activated, the effects of the ring oscillator may be observed directly from the power trace or by performing frequency analysis on the power trace.

Additionally, because ring oscillators are widely used in circuits for hardware trojan detection, it may be possible for an adversary at the manufacturing level to modify the mask of the circuit in such a way that portions of on-chip ring oscillators may be used for the payload of the HTH. In this case, the HTH may be mistakenly considered to be a part of the original circuit and may go undetected.

In this thesis, a ring oscillator has been used a the payload for the HTH. Because ring oscillators are normally of two kinds, two HTH payloads have been implemented and analyzed. One implementation consists of an INV chain and the other implementation consists of a NAND chain, along with an AND gate that enables/disables the HTH based on the logic value of the trojan trigger. The gate-level schematic of the Trojan payload with an inverter chain is as shown in in Fig 4.3.

4.3.1 Effects of RO chain length and type

A ring oscillator chain can vary from a single INV/NAND gate to many INV/NAND gates, provided that the number of inversions is odd. The implementations of an INV-based RO and a NAND-based RO with chain size 3 are as shown in Fig 4.4 and 4.5 respectively. In this section, the effects of the size and type of ring oscillator are analyzed.

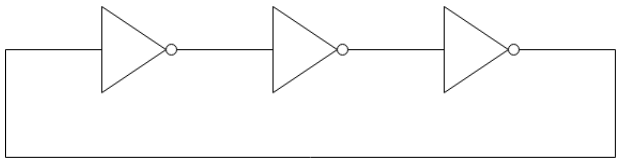

Figure 4.4: INV-based ring oscillator with chain length 3

4.3.2 Effect on Power

The hardware trojan consists of a trigger and a payload, wherein the payload is the ring oscillator circuit. In this experiment, we observe the effect of RO chain length on the power consumption by measuring the transient power of the ring oscillator payload through SPICE

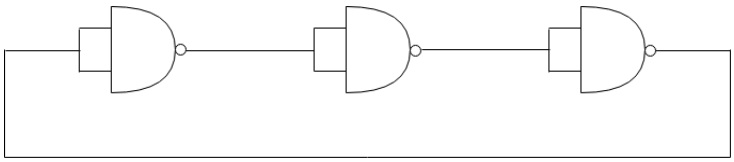

Figure 4.5: NAND-based ring oscillator with chain length 3

simulation. This experiment is performed to make conclusions on which type of RO would be more/less detectable in the power trace. The result of the experiment is as shown below. Fig 4.6 shows the power trace when a INV-based ring oscillator of chain size 7 is excited.

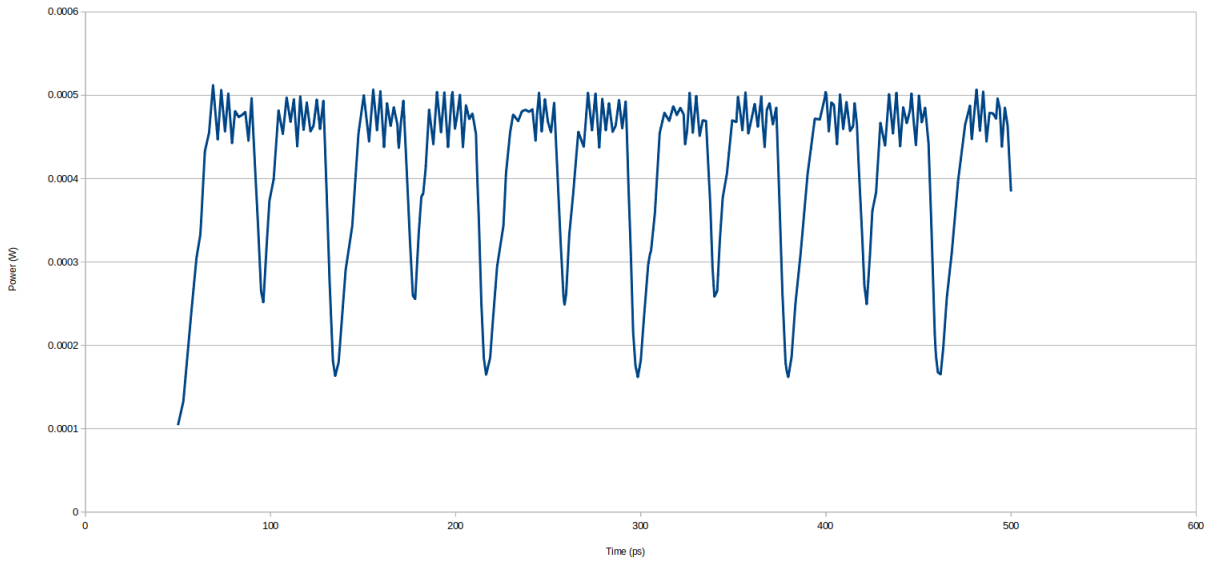

Figure 4.6: Power trace of INV-based RO with chain length of 7

Because a ring oscillator requires odd numbers of inverters, for a comparison in terms of power consumption, we simulated ring oscillators with 3, 7, and 13 inverters, and 3, 7 and 13 NAND gates. All the ring oscillators are enabled, and we evaluated the effect of size and type of RO on the power trace.

The power consumption of an INV-based RO with chain size 3, 7 and 13 is as shown in

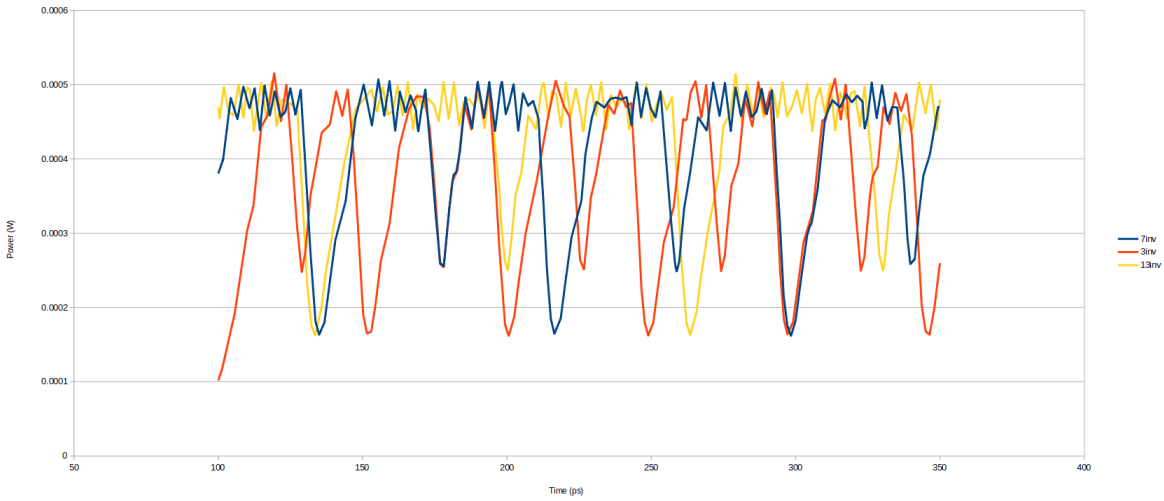

Figure 4.7: Power trace of 3, 7 & 13 INV-based ROs

Fig 4.7, and that of a NAND-based RO with chain size 3, 7 and 13 is as shown in Fig 4.8. Here, the legend is named according to the size and type of RO chain. For example, "7inv" would represent a INV-based RO with a chain length of 7. The legends in both these graphs are used to indicate different chain types. From the graphs, there are two main observations that can be made.

Firstly, the maximum power consumption for the INV-based ROs is approximately 0.513mW, regardless of chain length. In contrast, the maximum power consumption of NAND-based ROs is approximately 0.7105mW.

Secondly, although the maximum power consumption may be the same, the 3 traces for both the INV-based and the NAND-based ROs do not match exactly with respect to time. This is because the chain size of the ROs determine at what point of time the main AND gate that connects the rest of the RO to the trigger circuit, as shown in Fig 4.3, switches. For example, in Fig 4.7 the power trace of the RO with an INV chain length of 7 has the AND gate switch at about 140ns, after which it is followed by seven peaks at a power level of 0.5mW that indicate each of the seven inverters switching. This is again followed by the switching of the AND gate, which corresponds to maximum switching.

Hence, if an adversary tries to obtain data on whether or not the RO-based HTH is enabled, they could perform frequency analysis on the power trace based on the individual

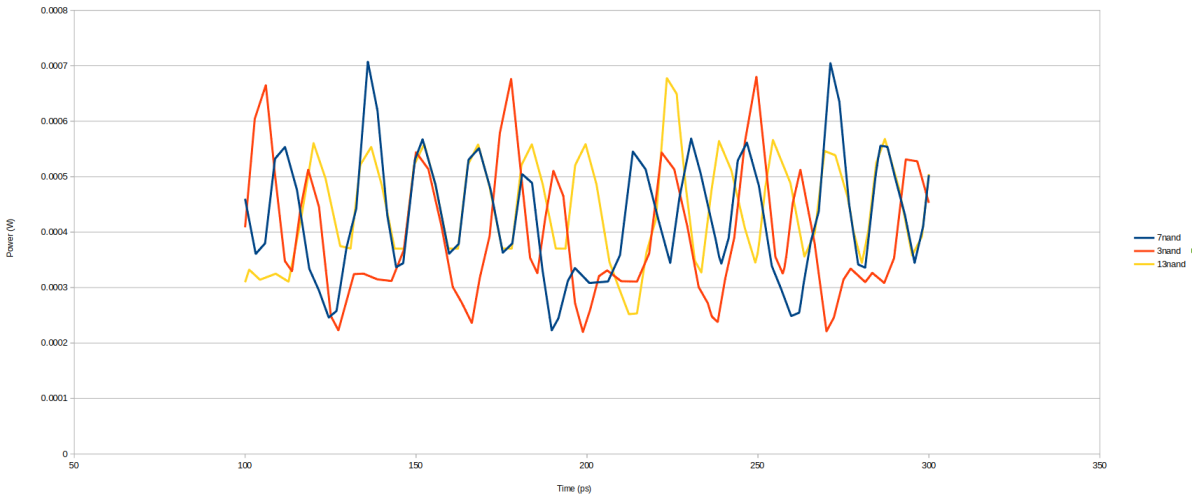

Figure 4.8: Power trace of 3, 7 & 13 NAND-based ROs

frequency components as seen in Fig 4.7. In fact, each trace in Fig 4.7 and Fig 4.8 is made up of multiple frequencies. This can be attributed to two factors. The first factor is, as mentioned before, the switching of the main AND gate. This is because, the longer the RO chain, the more the time between two consecutive switching activities of the AND gate. The second factor is that switching from logic 1 to logic 0 will consume a different amount of power compared to when switching from logic 0 to logic 1. It should however be noted that if parasitic extraction was performed and the circuit was then modelled using real capacitance values, the power trace of the ring oscillators may look more damped.

If an adversary tried to implement the RO as payload to the HTH, they may use an INV-based RO instead of a NAND-based RO because of decreased power consumption. This would mean that if the HTH is triggered, a NAND-based RO would be more visible in the power trace when compared with an INV-based RO. Because one of the goals of this thesis is to be able to extract data through frequency analysis of the power trace, the actual power consumption of the RO-based HTH does not matter as much as long as we are able to obtain correct data by using filters.

4.4 Power Analysis

For this thesis, the two ISCAS circuits C432 [31] and C499 [32] are used. The DPA-mitigation technique, SDDL, as well as the RO-based HTH, are implemented on these circuits. The circuits were first synthesized to use the NanGate 45nm library. The synthesis was performed to use only the AND, OR and INV gates as this would make implementing SDDL easier. After synthesis, SDDL logic was added to the synthesized circuit through cell substitution using Perl. This implies that for every synthesized AND gate, substitution would add two more AND gates and one OR gate and make the gate connections according to the SDDL circuit diagram. Accordingly, for every synthesized OR gate, substitution would add two AND gates and one OR gate and make the gate connections according to the SDDL circuit diagram.

An ideal implementation of SDDL should make the power consumption due to switching activity equal throughout the power trace, regardless of the operation being performed. However, due to the inclusion of duplicate inverted functions in our implementation of SDDL, this ideal condition is not possible to achieve. This is because of the structure of SDDL, where inverted outputs are formed by the inclusion of complementary gates (e.g. OR is complementary to NAND). In order for SDDL to be optimized, any pre-existing inverted functions should be replaced by inverted functions obtained through the implementation of the relevant complementary gate. This, however, does not affect or modify the effect of the HTH in any way.

4.4.1 Observation of Point P

As one of the goals of this thesis is to demonstrate that the logic value of any arbitrary point P in the circuit can be extracted through frequency analysis of the power side-channel, the RO-based HTH is implemented in C432 and C499. For this thesis, an arbitrary point in each circuit was selected as point P. The HTH can be easily modified to be able to observe any point in the circuit that the attacker desires by placing the HTH at that point in the circuit.

During the experimentation process, the power consumption of four implementations of the circuits are performed, and they are as follows:

1. Original circuit
2. Original circuit with HTH activated
3. Original circuit with DPA-mitigation activated
4. Original circuit with DPA-mitigation and HTH activated

4.4.1.1 Circuit 1

Our first circuit is C432. C432 is an ISCAS-85 combinational circuit that works as a 27-channel interrupt controller. It consists of four 9-bit inputs, 3 single-bit outputs and one 4-bit output. The input sequence used was obtained from the ISCAS HLM webpage for functional tests and it is such that C432 would have to perform various functions like enabling or disabling interrupt requests based on the priority assigned to the ports [31].

Fig 4.9 shows the power trace for C432 without any modifications, and Fig 4.11 shows the power trace of C432 without DPA-mitigation, but with the HTH inserted. In this case, the effect of the ring oscillator can be seen clearly from Fig 4.11 at intervals such as 150ns to 200ns, where there is increased power consumption in certain areas of the power trace where there would ideally be no switching activity. This increase in power consumption is because the HTH trigger is turned on, and the point P in C432 being observed is at a logic high. This becomes very clear when the power trace of C432 with HTH enabled is compared with the logic levels of point P (shown in Fig 4.10). The increased power consumption ranges from 0.2mW to 0.5mW, and is in accordance with the power consumption we had measured for the INV-based RO from Fig 4.6.

If the adversary is able to obtain the power trace but is not able to tell what the value of point P at any given point of time is through the power trace directly, the adversary can choose to filter out low-frequency signals from the power trace. For frequency analysis, the instantaneous values for power consumption were imported into MATLAB, after which the entire trace was passed through a High-Pass Filter (HPF). The HPF blocks the lower frequencies, and allows only the high frequencies to pass. Because the RO is the only high-frequency component in this circuit implementation, we can directly use an HPF instead of a Band-Pass Filter (BPF). As a result, we obtained the high-pass filtered trace as shown in

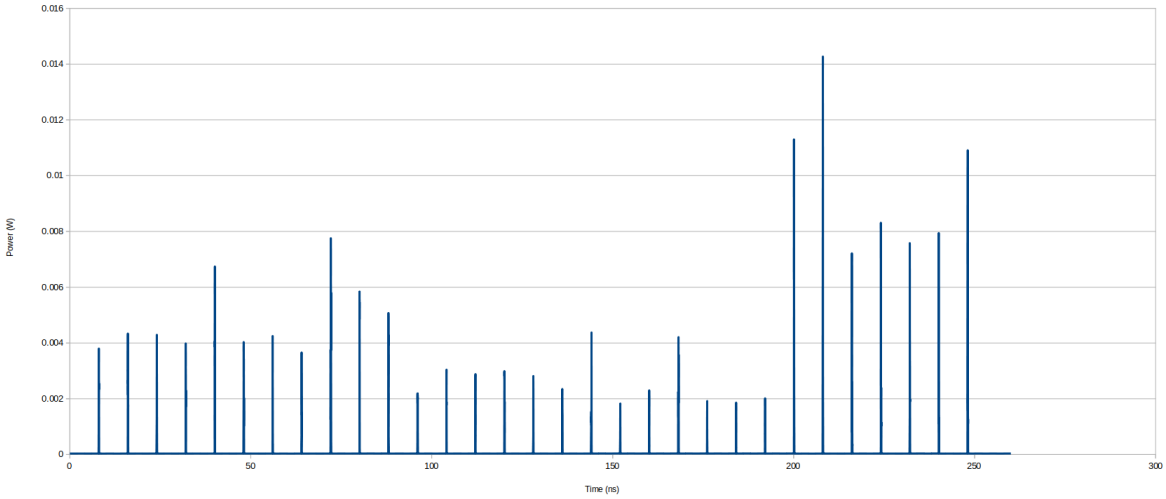

Figure 4.9: Power trace of C432

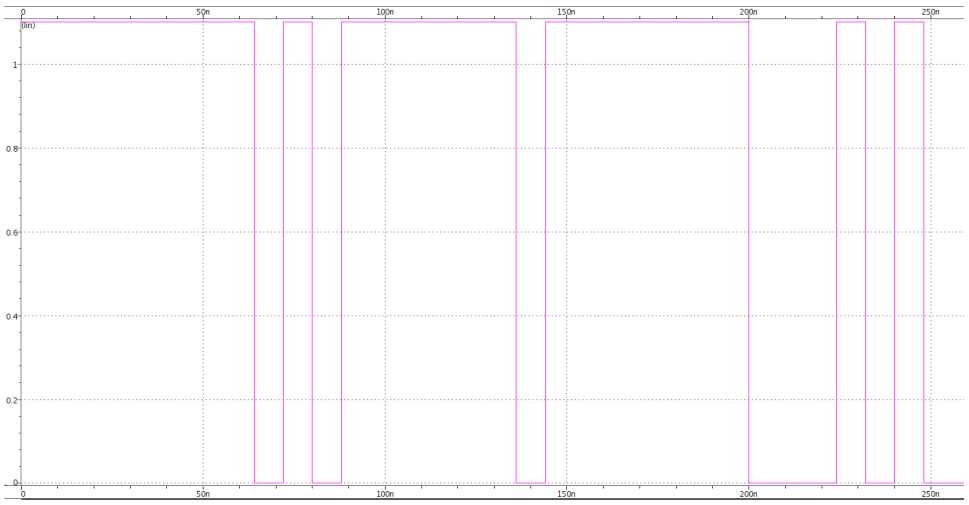

Figure 4.10: Waveform of point P being observed in C432

Fig 4.12. The correlation between Fig 4.10 and Fig 4.12 can be seen clearly as the filtered signal oscillates whenever point P is at a logic high, throughout the trace.

To ascertain the logic values represented by the filtered signal match with that of point P, a bitstream of the logic value at every 4ns was created. The actual value of P shown in

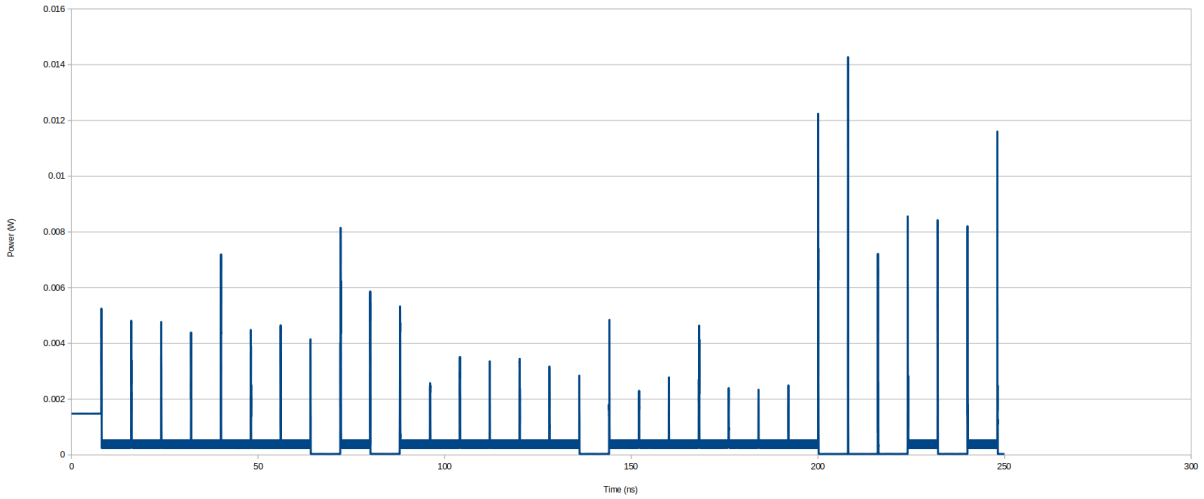

Figure 4.11: Power trace of C432 with HTH activated

Fig 4.10, has the following values:

1111111111111111001100111111111111001111111111111100000011001100

For point P being observed by the HTH inserted in C432, we choose to consider a logic value of 0 for the first 8ns as the circuit is still stabilizing here. The bitstream generated for the same is:

0011111111111111001100111111111111001111111111111100000011001100

Both bitstreams are a perfect match for the entire time period other than the first 8ns. This would mean that if the circuit is not protected by any DPA-mitigation techniques, the operation of the RO can be observed clearly through analysis of the power trace, given that the adversary has the required high-precision instruments to obtain good data.

Fig 4.13 shows the power trace for C432 with SDDL implemented for DPA-mitigation. Fig 4.14 shows the power trace of C432 with DPA-mitigation, and the HTH is enabled. When comparing this power trace to that of C432 with DPA-mitigation but without HTH enabled, the adversary may not always be able to tell when the RO is oscillating and when it is not. Thus, an adversary may pass the signals through an HPF to obtain the graph in

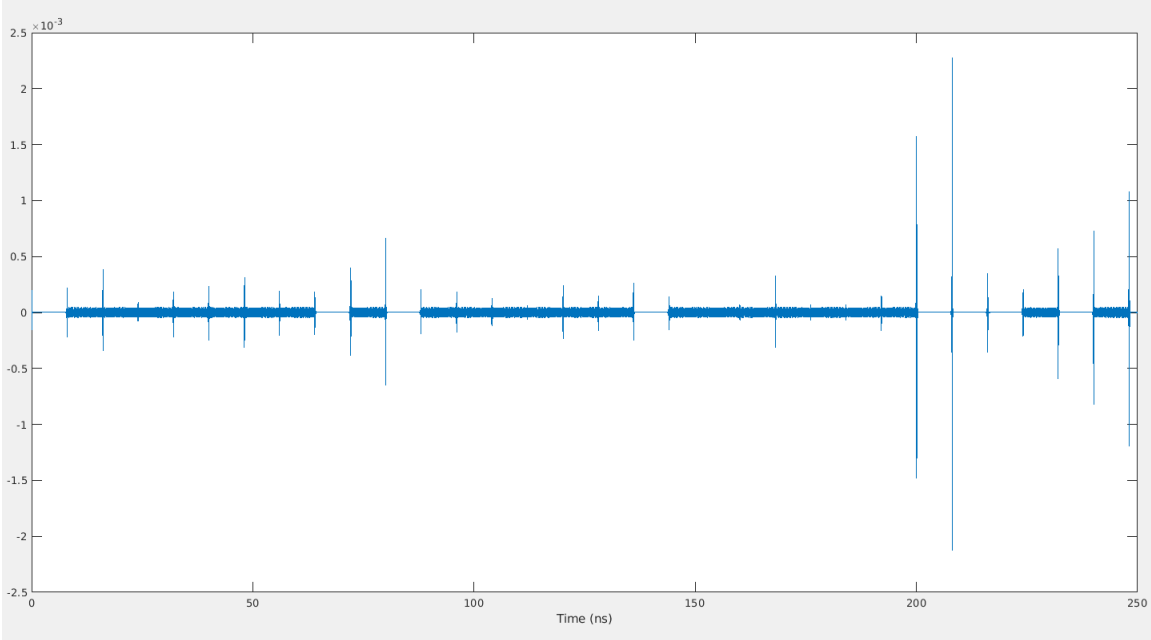

Figure 4.12: Frequency analysis of C432 with HTH activated

Fig. 4.15. The filtered signal can be visually observed to be a good representation of the point P that the HTH was leaking information on (Fig 4.10).

For the value of point P being observed by the HTH inserted in C432 with DPA-mitigation, the bitstream generated has the following values:

11111111111111110011001111111111110011111111111111000000011001100

The value of the bitstream matches with the bitstream generated for point P 100% of the time. This means that information on the value of point P is successfully being leaked by the HTH.

The effect of the ring oscillator can be seen in the power traces of Fig 4.13 and 4.14 due to minor changes to the amount of power consumption. For example, for the time period of 5ns to 50ns, the maximum amount of power consumption for the DPA-mitigated C432 with the HTH is about 68mW, and the minimum amount of power consumption is about 1.655mW. In contrast, for the time period of 5ns to 50ns, the minimum amount of power consumption for DPA-mitigated C432 without the HTH is about 1.49mW. But it is important to note

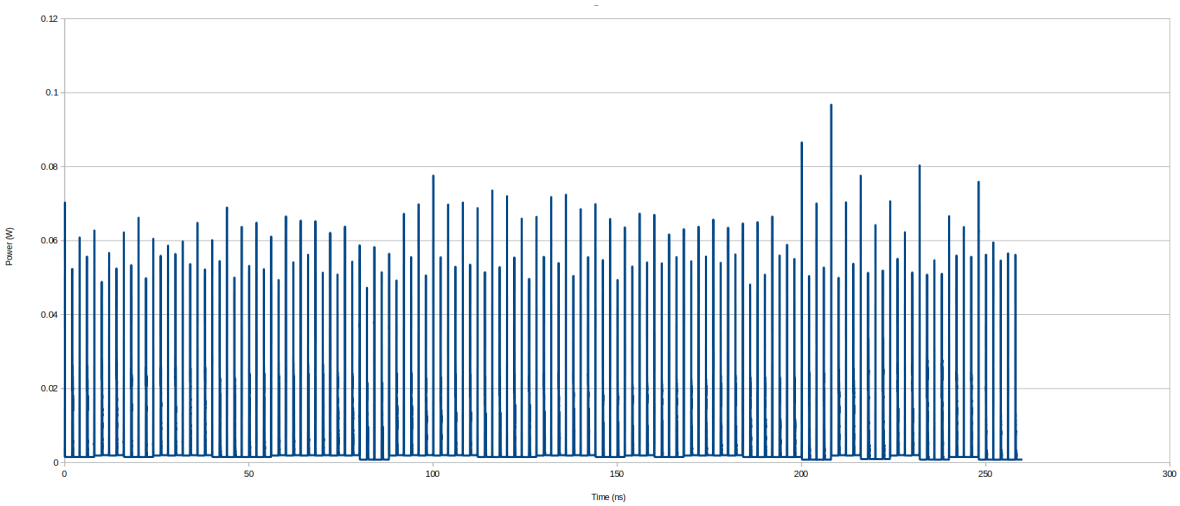

Figure 4.13: Power trace of C432 with SDDL implemented

that since this is a highly zoomed-in version of the trace, and since the difference in power consumption is minor, this increase in power consumption might be treated as noise. Thus, the HTH may or may not be detected during parametric test, considering that the HTH is activated. The precision and range of the test equipment will also play a big role in whether or not this HTH can be detected directly from the power trace.

In order to ascertain that point P can be observed by implementing this HTH, frequency analysis was performed. For C432, the actual value of the point P is as shown in Fig 4.10. The value of point P as deduced by frequency analysis is as shown in Fig 4.15. In this figure, the thick blue line represents times when point P is at a logic 1. As can be seen from the waveforms, we have successfully been able to observe the logic value of point P by passing the power trace through a filter.

Once again, to ascertain whether the logic values represented by the filtered signal in Fig. 4.15 match the true value of point P, a bitstream of the logic value at every 4ns was created. For the value of P observed in Fig 4.10, the bitstream generated has the following values:

1111111111111111001100111111111111001111111111111100000011001100

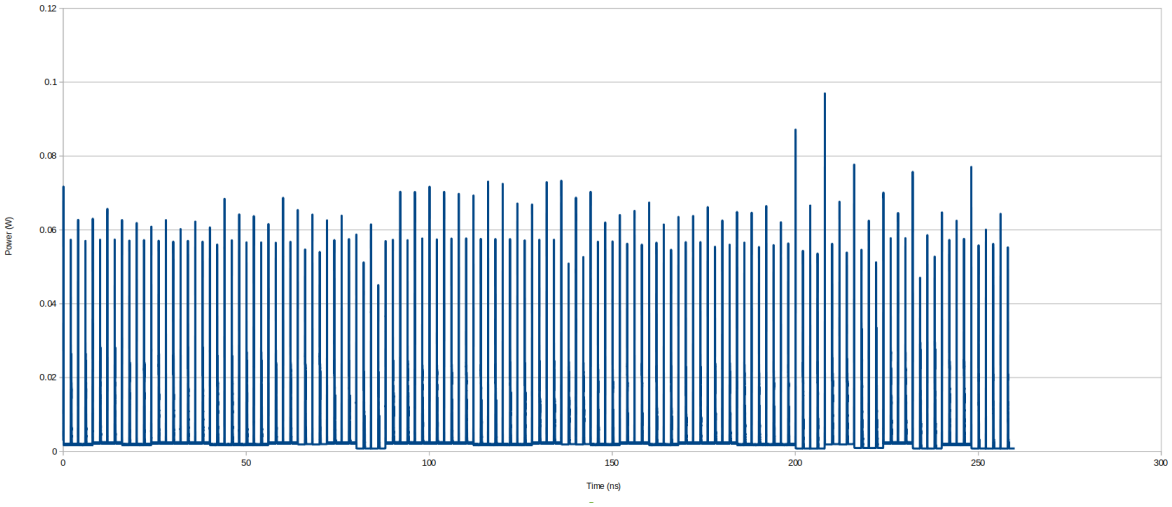

Figure 4.14: Power trace of C432 with SDDL and HTH activated

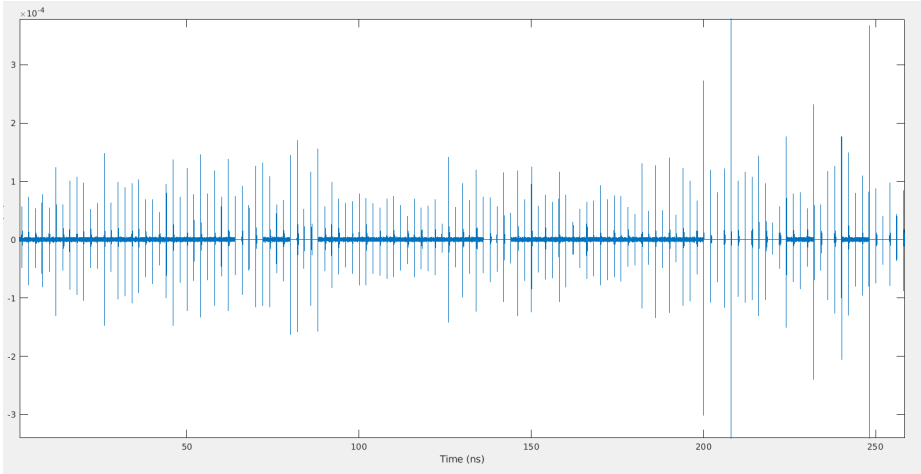

Figure 4.15: Frequency analysis of C432 with SDDL and HTH activated

This bitstream matches exactly with the bitstream for point P, thus implying that information on point P can be successfully be leaked in C432 even when it employs SDDL as a DPA-mitigation technique.

4.4.1.2 Circuit 2

The second circuit on which we experimented is C499. C499 is an ISCAS-85 combinational 32-bit single-error-correcting circuit. It consists of one 32-bit input, one 8-bit input,

one single-bit input and one 32-bit output. The input sequence used was obtained from the ISCAS HLM webpage for functional tests [32].

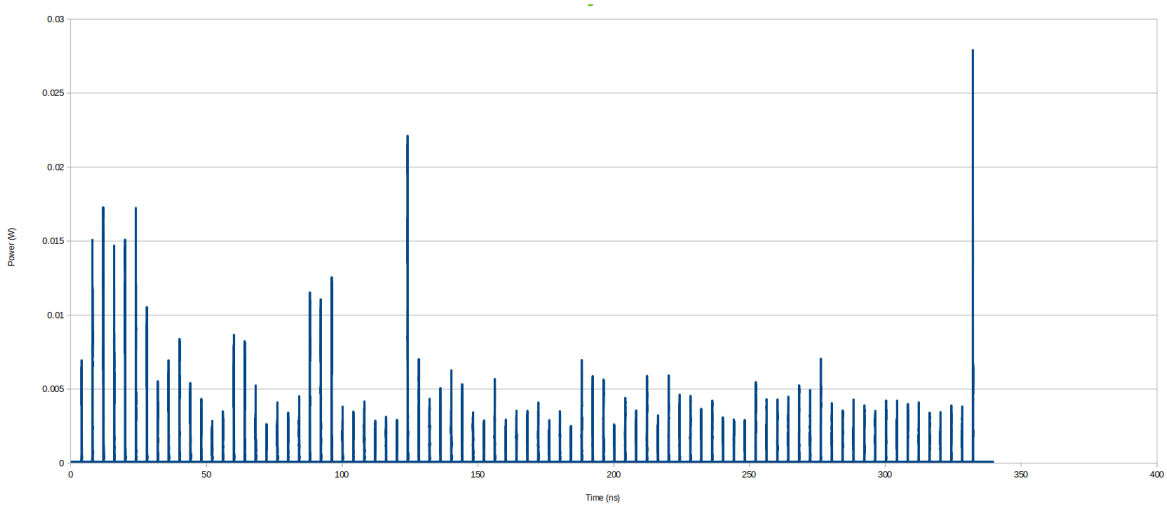

Figure 4.16: Power trace of C499

Fig 4.16 shows the power trace of C499 without DPA-mitigation and without the HTH activated. Fig 4.17 shows the logic values of the point P being observed by the HTH inserted in C499 for the entire time period of simulation. Additionally, for analysis purposes, we assign bit values for every 4ns in the trace for point P to be able to observe it closely and compare it to the modified circuits. The bit value for point P is given as:

0000010000000000000000000000100111111111111111111111111111110111111111111111111111

Fig 4.18 shows the power trace of C499 without any DPA-mitigation, but with the HTH activated. The effects of the ring oscillator-based HTH may be visible to the attacker through visual inspection as the power trace in Fig 4.18 clearly shows that the baseline power consumption is raised by approximately 0.5mW when point P is equal to logic 1. Based on

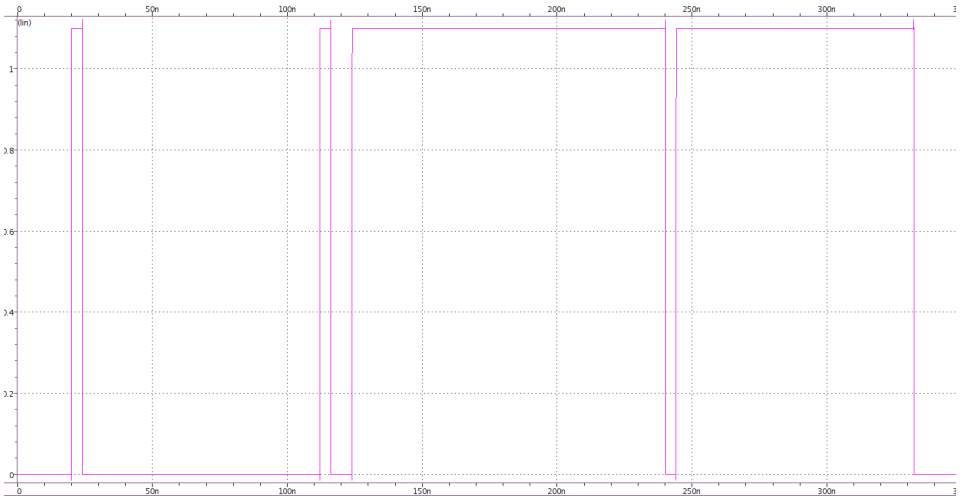

Figure 4.17: Waveform of point P being observed in C499

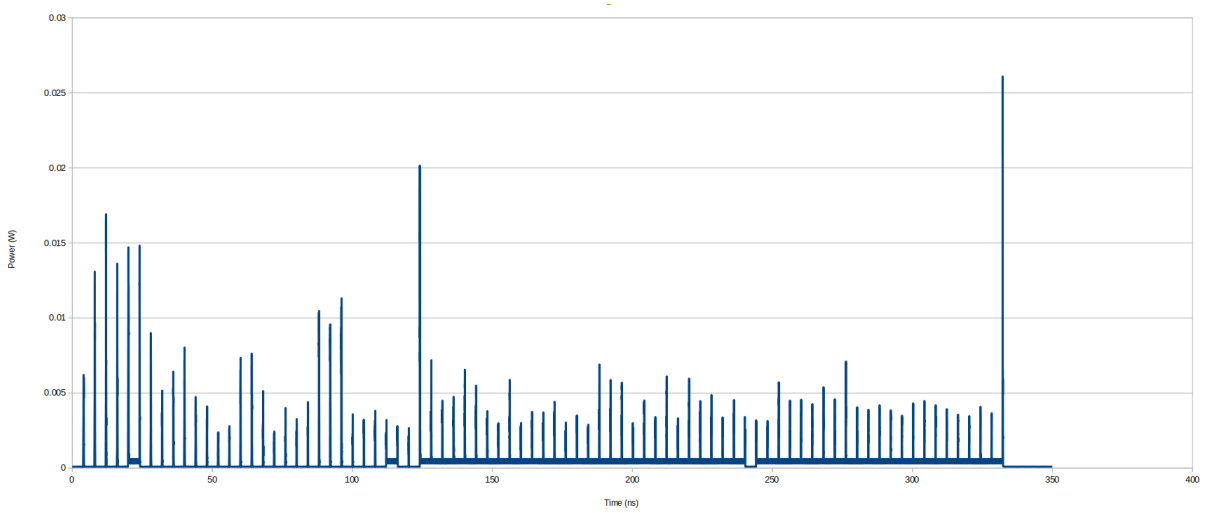

Figure 4.18: Power trace of C499 with HTH activated

visual analysis, the value of point P as leaked by the HTH is encoded in a bitstream as:

000001000000000000000000000010011111111111111111111111111111011111111111111111111111

The value of the two bitstreams match 100%, which means that the RO-based HTH successfully leaks information on point P through the power trace. However, considering

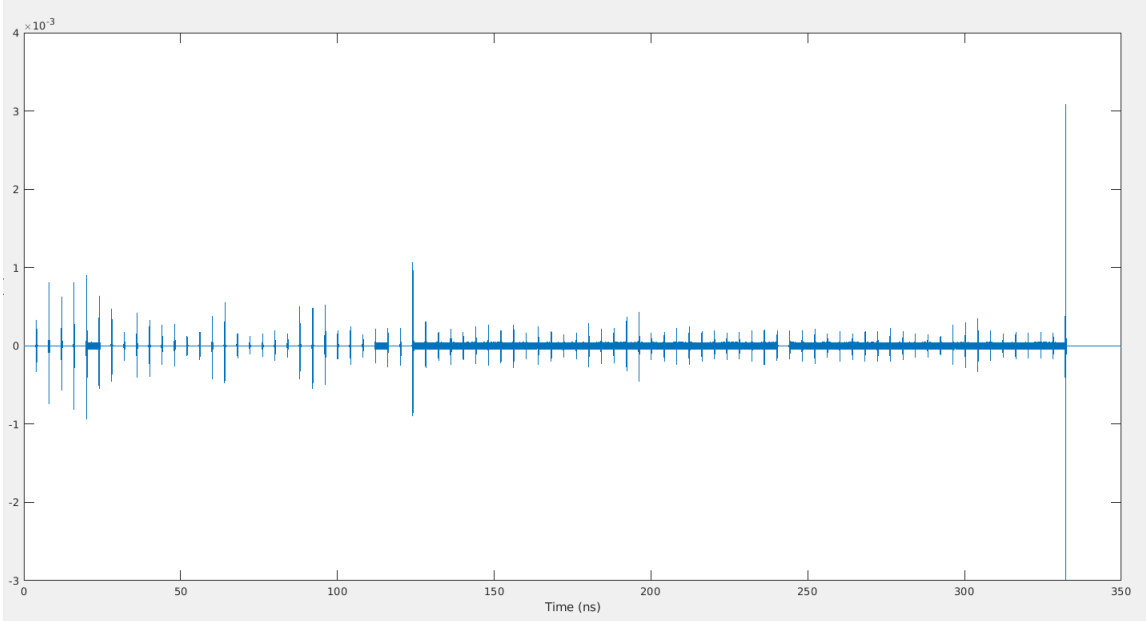

Figure 4.19: Frequency analysis of C499 with HTH activated

that the effect of the ring oscillator may not be entirely clear through visual analysis, the attacker may decide to use an HPF on the power trace to filter out all the low frequency signals. The filtered trace for C499 is as shown in Fig 4.19. From the trace, it becomes very clear as to when we are able to observe the high frequency signals and when there are no high frequency signals. The high-frequency signals here refer to the oscillation of the ring oscillator, and when we assign logic values to the filtered signal, we get the following bitstream:

00000100000000000000000000010011111111111111111111111111111011111111111111111111

When compared to the point P being observed by the HTH in C499, we can effectively say that information on point P is successfully leaked by using this HTH as the values of the bitstream match completely with the bitstream generated for point P.

Fig 4.16 shows the power trace of the original C499 circuit and Fig 4.20 shows the power trace of C499 with DPA-mitigation but without the HTH enabled, and Fig 4.21 shows the power trace of C499 with DPA-mitigation as well as the HTH activated. In this case, the

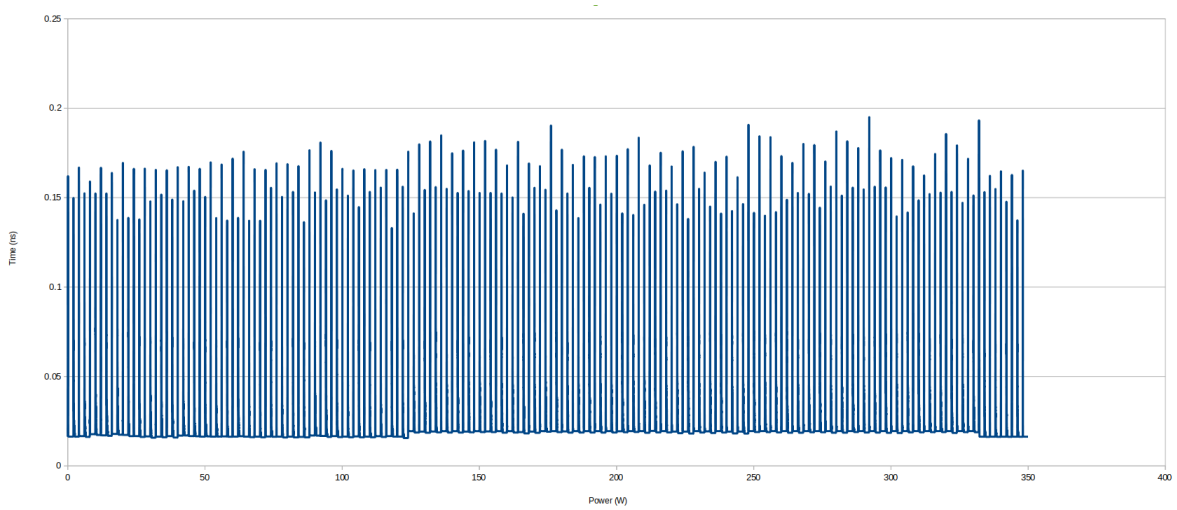

Figure 4.20: Power trace of C499 with SDDL implemented

effect of the ring oscillator may not be completely visible by means of simple visual inspection. Hence, an adversary may try to pass the trace through a high-pass filter to isolate the high frequency signals and thus be able to identify when the ring oscillator is enabled and when it is not. After passing the power trace through an HPF in MATLAB, the resultant signal is as shown in Fig 4.22. Disregarding the background noise, an adversary may try to assign logic values to the filtered signal. On the basis of logic value for each time period (4ns), we get the following bitstream:

0000010000000000000000000001001111111111111111111111111111101111111111111111111111

The value of the bitstream matches completely with the bitstream generated for point P. Hence, we can conclude that the adversary is able to leak and observe the logic value of point P by means of passing the power trace through an HPF and analyzing the resultant graph.

Hence, this would mean that although the circuit may be protected by a DPA-mitigation technique like SDDL, and although the effects of the HTH may not be directly observable from the power trace, the operation of the RO can be observed clearly by the adversary through frequency analysis of the power trace, given that the adversary has the required

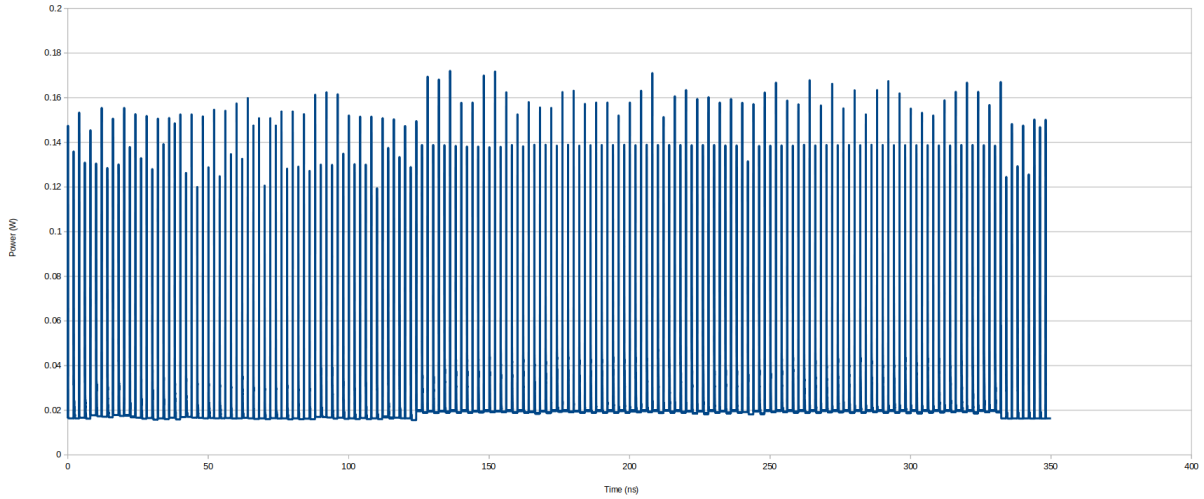

Figure 4.21: Power trace of C499 with SDDL and HTH activated

high-precision instruments to obtain good data.

4.5 Frequency obfuscation

Because it appears to be possible to detect the information being leaked by the Trojan horse even in the presence of SDDL, an additional circuit was developed to create additional switching activity and mitigate the Trojan effect. In this thesis, we are introducing a type of frequency obfuscation technique which will prevent an attacker from obtaining leaked information through frequency analysis of the power trace. This technique works by obfuscating the power trace to such an extent that high-frequency information, relayed on the power trace by constant and high-speed switching, cannot be leaked even if there is an elongation of the clock period and/or an increase in the time interval used to apply new inputs. The circuit design for the DPA-mitigation technique is as shown in Fig 4.23.

The circuit shown in Fig 4.23 comprises two MUXs, one NAND and one XOR gate, and includes combinational feedback. The idea behind implementing this technique is that the value of x_ns will keep toggling regardless of whether input x is in the same logic state as it was previously or not. The value of x may correspond to a point in the circuit, an output of a Linear Feedback Shift Register (LFSR) or any other desired signal. Multiple copies of this obfuscation circuit may be added and attached to different signals until the desired level of

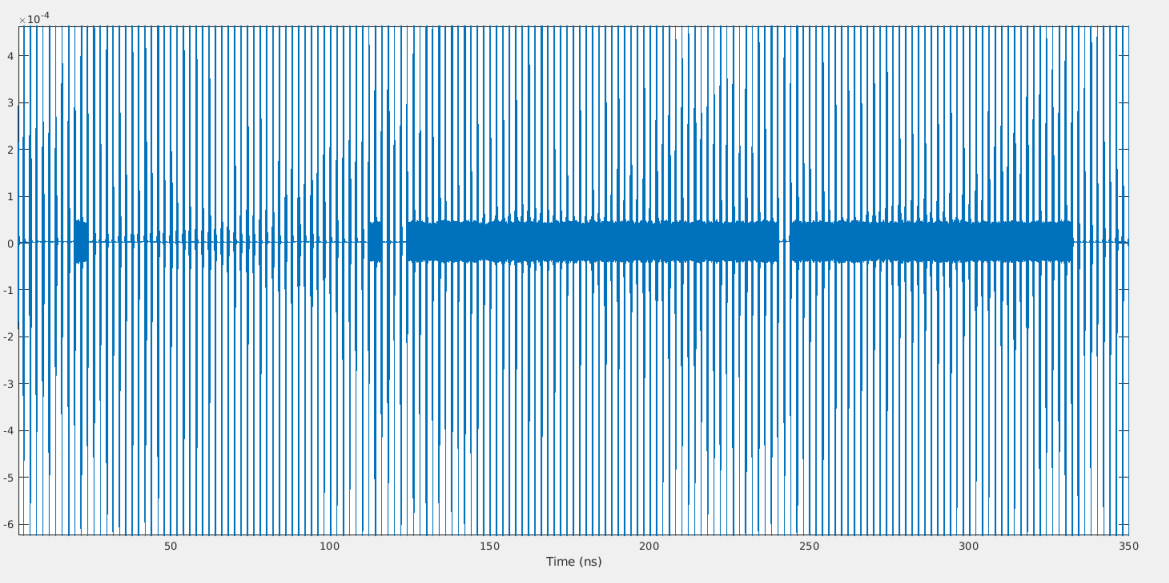

Figure 4.22: Frequency analysis of C499 with SDDL and HTH activated

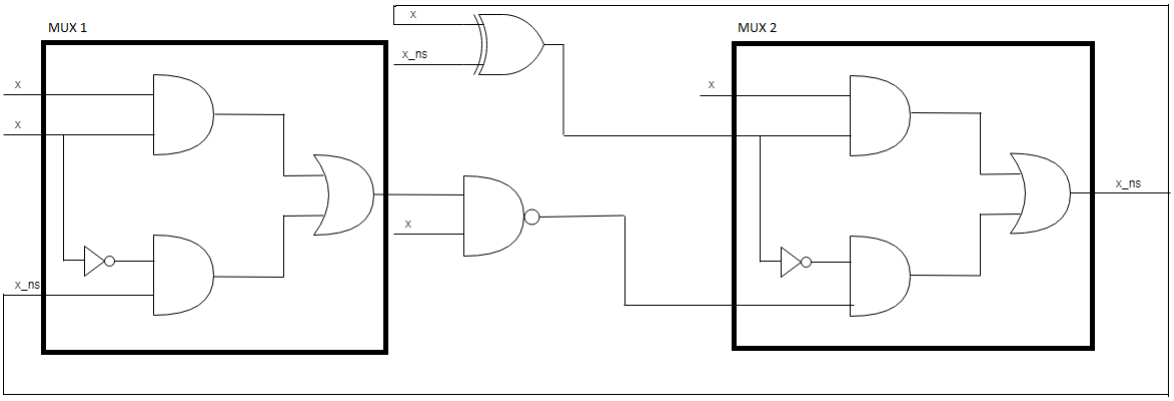

Figure 4.23: Implementation of DPA-mitigation

obfuscation has been reached. The truth table for this implementation is as shown in table 4.1.

Table 4.2 shows the switching activity for a single implementation of the proposed obfuscation technique. It shows the total switching activity within the circuit when the inputs change from one state to another for all of the input switching combinations. The three input state changes that cause a switching activity of 4 are 11 to 10, 10 to 11, and 00 to 11.

Table 4.1: Truth table for DPA-mitigation implementation

x	x_ns (Input side)	x_ns (Output side)
1	1	0
1	0	1
0	1	0
0	0	1

Table 4.2: Switching activity for state changes

x (Prev state)	x_ns (Prev state)	x (Next state)	x_ns (Next state)	Switching activity
1	1	1	0	4
1	1	0	0	5
1	0	1	1	4
1	0	0	1	6
0	1	1	0	6
0	1	0	0	6
0	0	0	1	6
0	0	1	1	4

Additionally, input state changes from 00 to 01, 01 to 10, 01 to 00, and 10 to 01 each cause 6 gates to switch. A change from 11 to 00 causes 5 gates to switch. Because the switching activity is not equal for all state changes, the power consumption will also hence not be equal for all the state changes. This means that if the adversary tried to manipulate the power trace in a way such that they subtract the estimated power consumption of the Trojan from the power consumption caused by the proposed obfuscation technique, they may still not be able to tell when the HTH is enabled versus when it is not. Additionally, this kind of comparative analysis on the side of the attacker may become even more difficult if more than one module of the proposed obfuscation technique is used.

Because the circuit for this obfuscation technique, as shown in Fig 4.23, has a combinational feedback loop, the circuit will continue to switch even after other circuit transitions have died out. This means that even if the ring oscillator-based Trojan is enabled while the proposed frequency obfuscation technique is active, due to the superposition of two separate frequencies in the overall power consumption, the adversary may find it more difficult to

extract information on the observed point P even if they pass the power trace through a bandpass filter. This is especially true because the number of gates switching due to the presence of the RO-based Trojan is within the normal variability of switching gates for this circuit.

Since this method is to be implemented by connecting "x" to wires and not gates, the wires can be chosen randomly from multiple sites in the circuit. These wires can be inputs or outputs to gates. The switching activity between the different state changes may be different, as shown in Table 4.2, but because x_ns keeps changing, the switching activity will be dynamic and more difficult to predict for an attacker a priori.

4.5.1 Implementation of proposed frequency obfuscation technique

The proposed frequency obfuscation technique is implemented on C432. The resulting power trace is shown in Fig 4.24. As can be seen, the power trace looks very different from that of the original trace for C432 in Fig 4.9 as it introduces random power consumption in the time between two consecutive input patterns applied to C432. Because of this, it may become more difficult for the adversary to be able to identify when the ring oscillator-based HTH is enabled and when it is not.

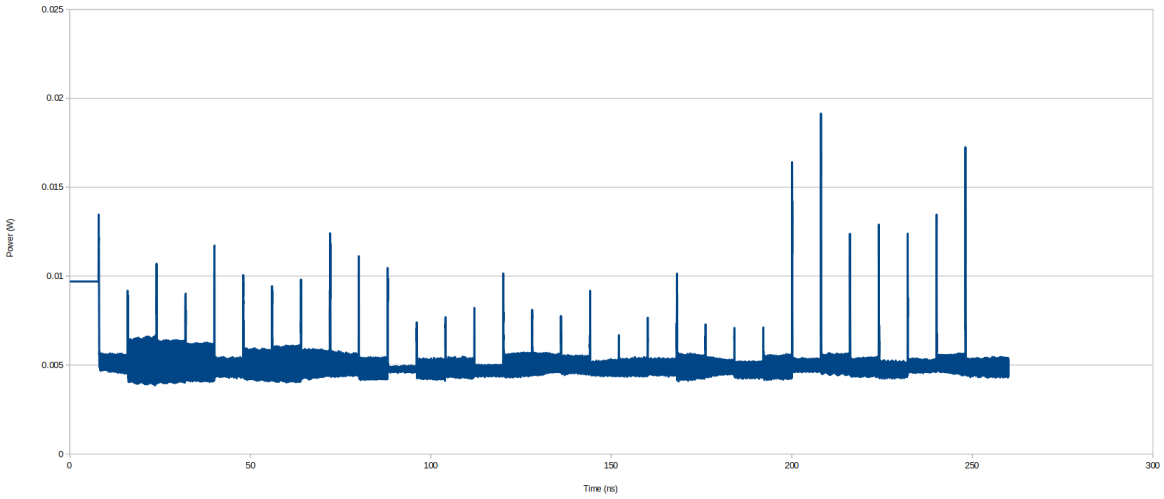

Figure 4.24: Power trace of C432 with proposed method

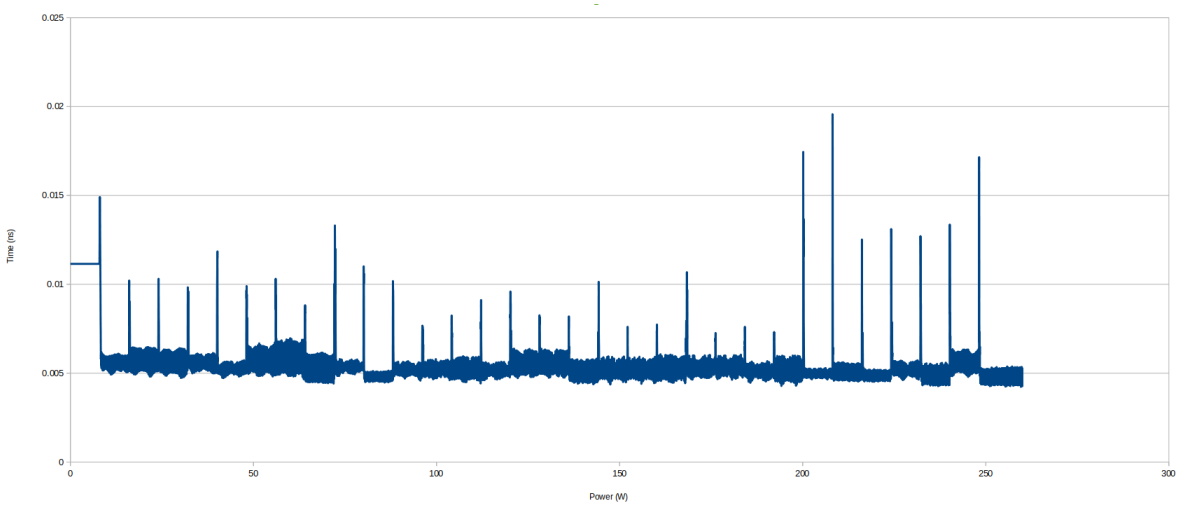

Figure 4.25: Power trace of C432 with proposed method with HTH activated

As a result of implementing this technique, high-frequency components are introduced to the power trace. Fig 4.25 shows the power trace of C432 with the proposed frequency obfuscation method while also having the HTH enabled. When compared to Fig 4.24, the power trace looks different in the periods between the input patterns applied of the original circuit, but it may be difficult for an adversary to be able to tell when the Trojan is enabled just by means of visual inspection. An adversary may try to then pass the power trace through a high-pass filter, and the result of passing the power trace through an HPF is as shown in Fig 4.26. Comparing this filtered signal to point P for C432 shown in Fig 4.10, it may still be difficult for the adversary to be able to tell the difference between the effects of the ring oscillator from the effects of random switching caused by the implementation of the proposed obfuscation technique.

If an adversary tries to extract the value of point P from the filtered signal, they may assign bit values to each time period (4ns). Because there are high-frequency components throughout the trace, an adversary may try to assign logic values to each time period in the trace based on the observed amplitude of the filtered signal. If the adversary considers the lower 50% of amplitude as logic 0 and the higher 50% as logic 1, the bitstream generated for the filtered trace is as follows:

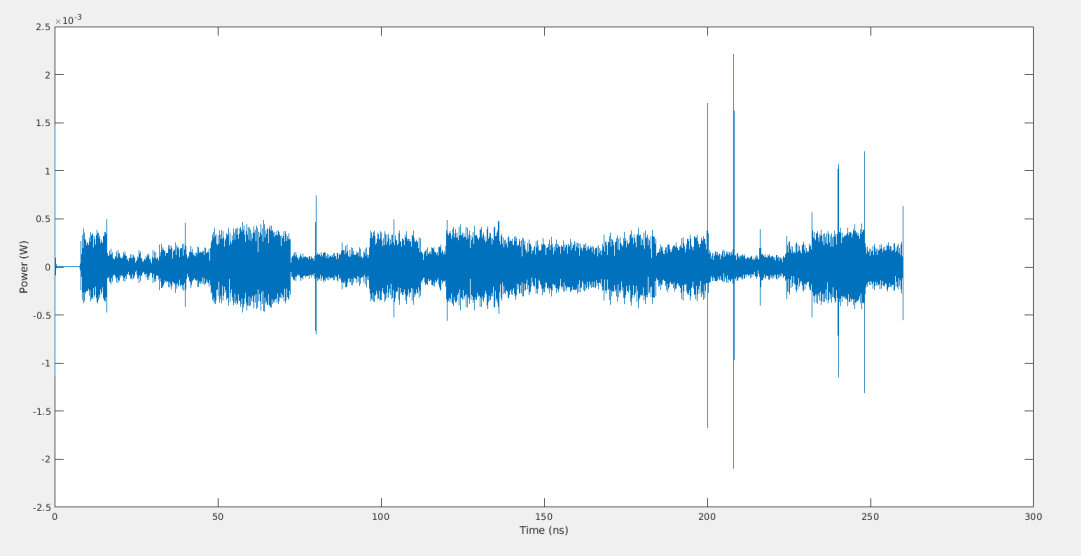

Figure 4.26: Filtered power trace of C432 with proposed method with HTH activated

00110000000011111100000011110011111111111111111111000000111111000

In contrast, the value of point P observed by the HTH in C432 was denoted by the bitstream:

11111111111111110011001111111111110011111111111111000000110011000

The bitstream obtained by analyzing the filtered signal in our experiment is the not same as the bitstream we obtained by observing the logic values of point P, and there is approximately 34% mismatch in values. Note that the amplitude of the signal will depend on the switching activity within the proposed frequency obfuscation technique, which is not constant and the variation is greater than the switching activity added by the RO-based HTH.

For further analysis, different threshold levels were applied to the trace and the bitstream for each was generated and compared with the actual values of point P. The chosen threshold is given as a percentage of the maximum power consumption in the filtered power trace. The generated bitstream assumed a value of 0 if the power level was below the chosen threshold,

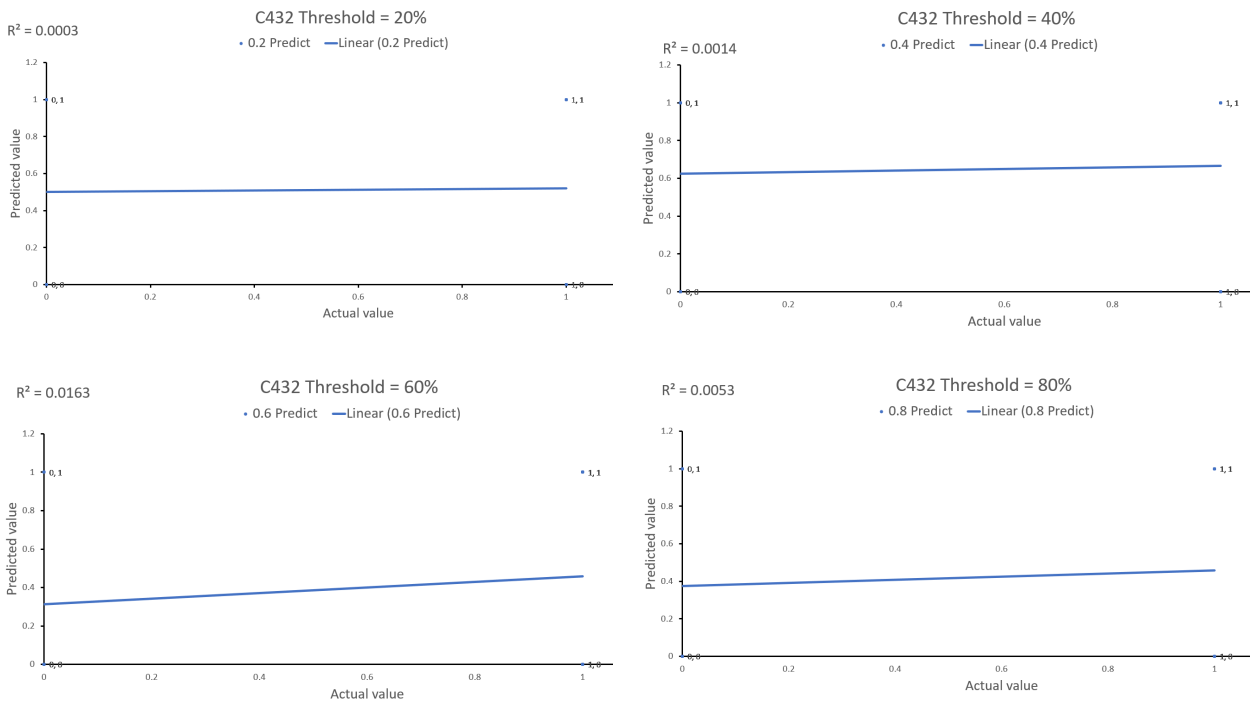

Figure 4.27: Regression for different thresholds in C432 with the proposed frequency obfuscation technique and HTH activated

and the value of 1 represented the power level was above the chosen threshold. A graph of the actual value vs the predicted value for point P was then plot and a trend line was generated for the same. Fig 4.27 shows the regression plots for different chosen thresholds. The trend line is a line that indicates the general tendency of data. For a correlation of 100%, the predicted and actual values for the $1's$ and $0's$ in the bitstream should match completely. If there is complete correlation, we would get a trend line with a positive slope that would angle at 45° from the x-axis. This would be so because the actual and predicted values for all the data points would be the same, and all the data points would thus either lie at (1,1) or (0,0) in this case. The value of R^2 is a representation of how closely fit the data is to the regression line. In the ideal situation, R^2 would have a value of 1. If the R^2 value is 1, it would mean that the attacker can successfully extract the correct values of the observed point P by using the chosen threshold.

By observing the values of regression for each of the thresholds in Fig 4.27, we can say

that there is little correlation between the actual value of point P and the predicted value of point P for any of the chosen thresholds. The R^2 values are very close to 0, which denotes little correlation and poorly fit data points. Comparing the five thresholds, a threshold of 50% was the most accurate in its representation of the logic values of point P with an error rate of about 34%. Thus, the amplitude of the filtered signal is more of a representation of when the frequency obfuscation technique has more switching activity, rather than the state of the ring oscillator-based HTH. Thus, the signal will be very noisy and will successfully mitigate this type of HTH.

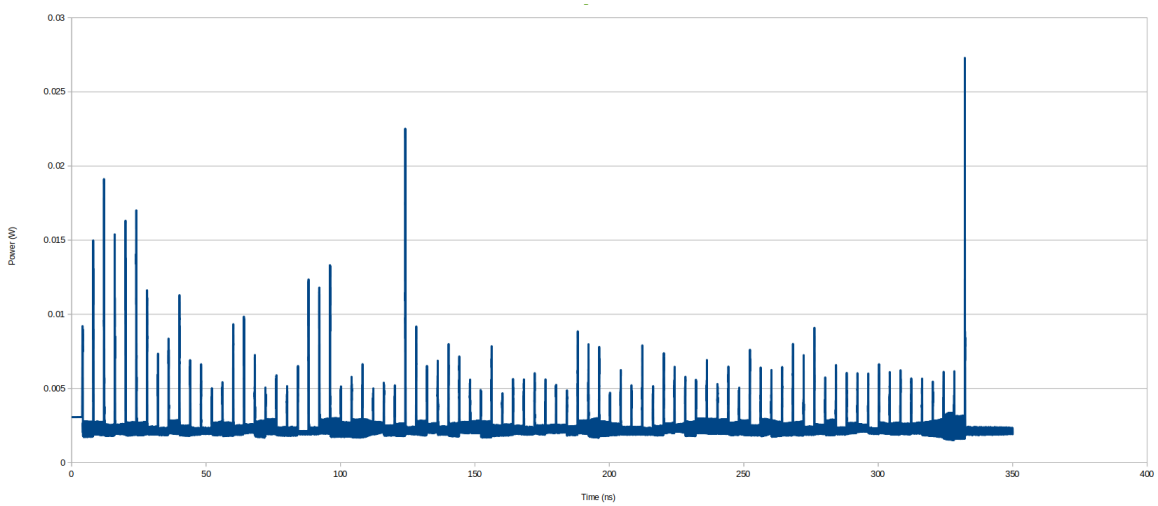

Figure 4.28: Power trace of C499 with proposed method

The same simulations were done using the second circuit, C499. Fig 4.28 shows the power trace of C499 with normal operation while implementing the frequency obfuscation technique. The input sequence is the same as the test inputs suggested by the HLM website. Fig 4.29 represents the power trace of C499 with the HTH enabled, with the frequency obfuscation technique implemented. Again, actual values of point P being observed here are as shown in Fig 4.17

Just by means of visual inspection, an adversary may not be able to tell whether or not

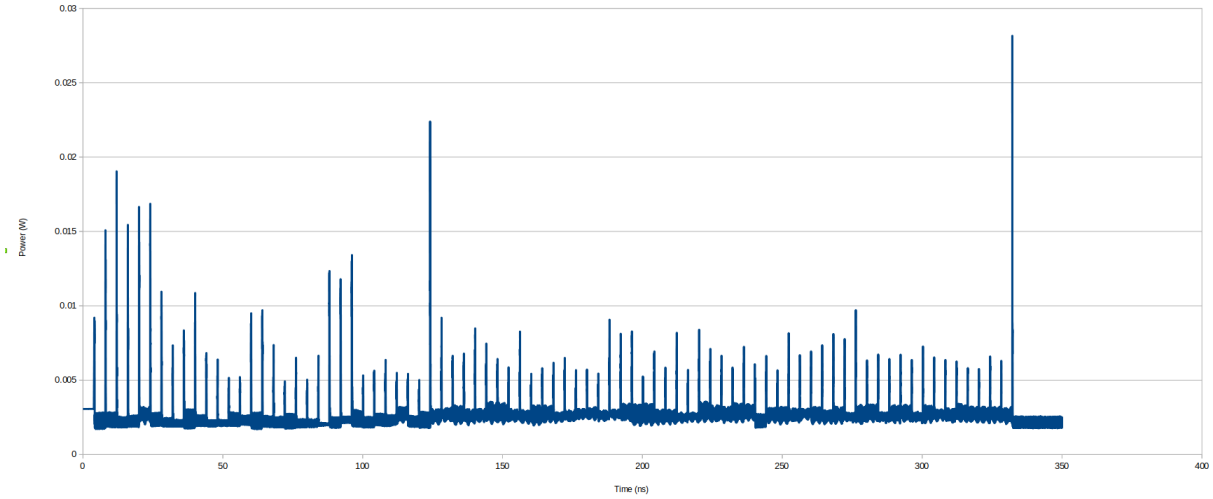

Figure 4.29: Power trace of C499 with proposed method with HTH activated

the ring oscillator is enabled as there are multiple frequency components in the power trace. Fig 4.30 shows the power trace of C499 with frequency obfuscation and HTH activated, passed through a high-pass filter. By means of visual inspection, it becomes clear that the correlation between point P for C499 and Fig 4.30 is very little.

For aiding our analysis, we assigned bit values to every 4ns, with a threshold of 50%, in the filtered signal. The bitstream generated is as follows:

0100111001100101111000000001111110111010100100001101100111000011001001111111011000 0

In contrast, the actual bitstream for point P is:

0000010000000000000000000000100111111111111111111111111111111011111111111111111111111

When compared to the original bitstream generated for point P, the new bitstream is incorrect approximately 44% of the time.

Fig 4.31 shows the trend line, along with the coefficient of determination for when different threshold values are assigned to generate the bitstream of the filtered signal. Here, the threshold is given as a percentage of the maximum amplitude of the filtered signal.

By observing the values of regression for each of the thresholds, we can say that there is

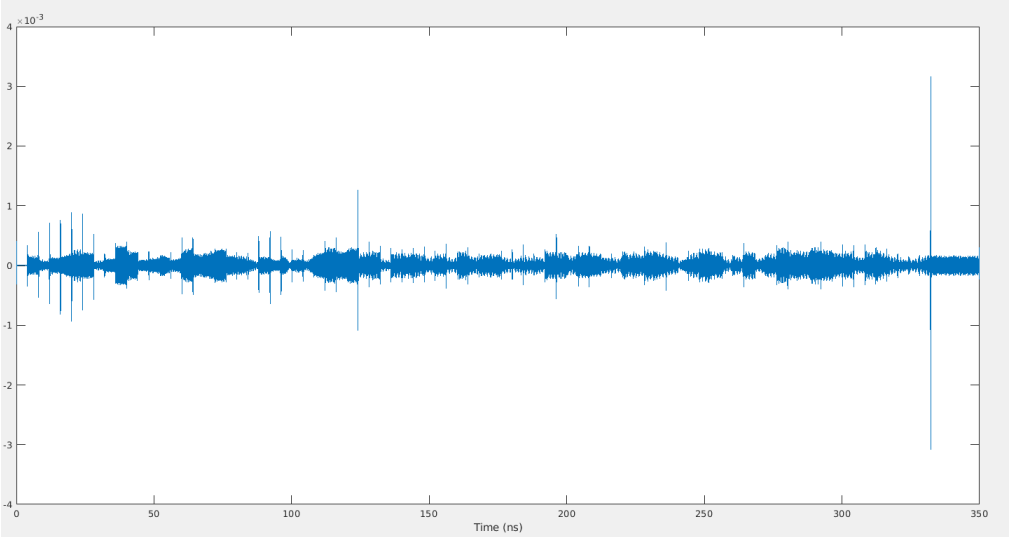

Figure 4.30: Filtered power trace of C499 with proposed method with HTH activated

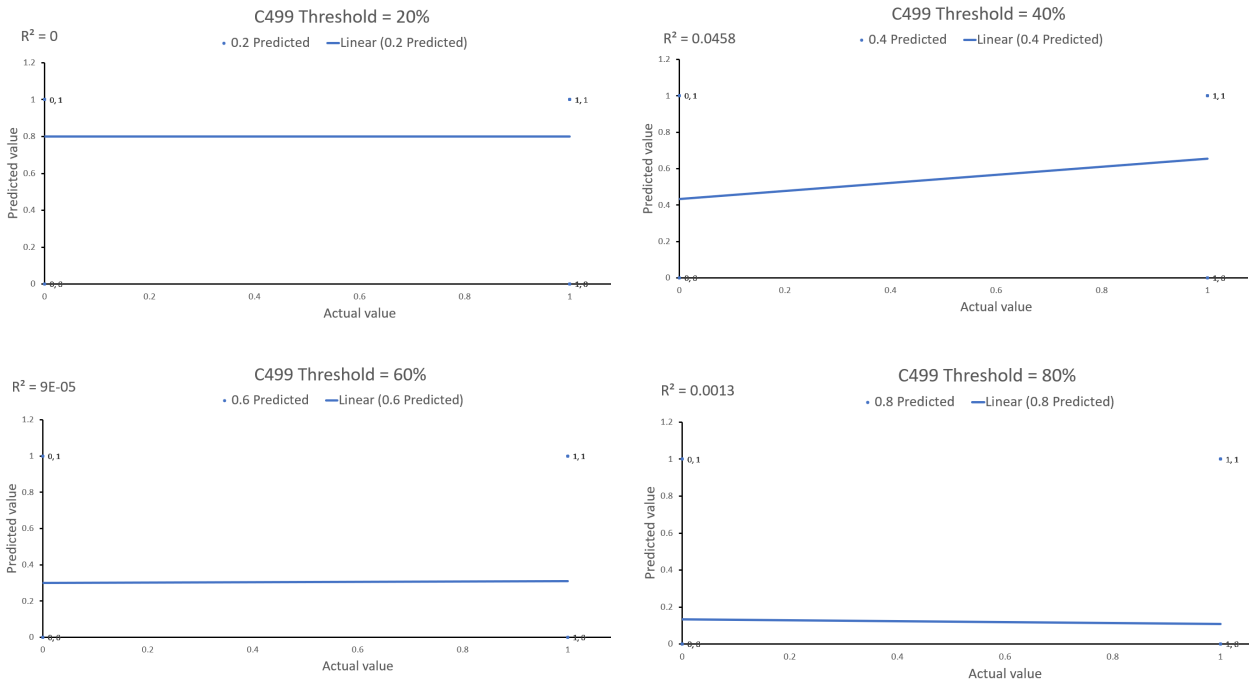

Figure 4.31: Regression for different thresholds in C499 with the proposed frequency obfuscation technique and HTH activated

little correlation between the actual value of point P, as determined by its bitstream, and the bitstream generated by assigning threshold values to the C499 circuit protected with the proposed frequency obfuscation technique. In fact, for a threshold of 20%, the regression coefficient has a value of 0. This means that there are as many values that represent the predicted value of point P correctly, as there are values that represent the value of point P incorrectly. In contrast, the R^2 value for a threshold of 40% is the best representation of point P as the R^2 value is the highest at 0.0458, and the error rate is 38.8%. However, this is still very poor correlation.

Hence, with the experiments for the two ISCAS circuits, C432 and C499, it becomes clear that by the implementation of the proposed frequency obfuscation technique makes it more difficult for an adversary to extract data from frequency analysis of the power trace.

Chapter 5

Conclusions and Future Work

5.1 Conclusion

In this thesis, we employed SDDL as a DPA-mitigation technique and showed that although DPA-mitigation may make it more difficult for adversaries to perform DPA attacks and extract internal circuit information, there is a work-around for this. The adversary may introduce the proposed HTH to leak information on the logic state of a point P in the circuit through the power trace. Additionally, this type of HTH need not be used in isolation. If an adversary is able to use this HTH to observe one or more key bits in an encryption circuit, it could also be used by attackers to make DPA attacks easier to implement by making it easier to generate the key hypothesis.

As shown in Chapter 4, extracting information regarding the logic value at point P by extracting the power trace and performing frequency analysis was implemented and was accurate in the data that it leaked. This type of HTH works because the oscillation frequency of the HTH is much faster than that of the clock. However, if there are other high-frequency components in the circuit, the HTH may not work so well.

In order to counter the effect of this HTH, we also suggested implementing a different type of technique for obscuring the power trace in such a way that any frequency-based analyses will not work. This method consisted of a circuit design that introduced greater variability into the power trace than that added by the Trojan. After simulation, we came to the conclusion that this method successfully made leakage of information regarding the value at point P through the frequency analysis of the power side channel more difficult. It did so by introducing other high-frequency components to the circuit.

Thus, in summary, we have implemented the HTH and verified its functionality and effects on the power trace. We also implemented a DPA-mitigation technique and verified

the HTH's functionality in the presence of that technique. We have also designed a frequency obscuring circuit that would help prevent the attacker from obtaining data regarding the value of point P through frequency analysis of the power trace.

5.2 Future Work

Future work will focus on further investigating the effects of the proposed power obscuring technique, along with analyzing how effective implementing it may be in terms of power, area and cost constraints. The effects of including multiple blocks of the proposed circuitry can be studied in detail in terms of how it affects the frequency spectrum. One can also implement this technique in tandem with the other pre-existing DPA-mitigation techniques to verify its effectiveness. Other techniques for prevention of information leakage through frequency side channel can also be investigated.

Additionally, the proposed frequency obscuring technique needs to be verified in more circuits, specifically different kinds of circuits. This will provide more information regarding the applicability of the proposed approach.

Future work will also include analyzing how the HTH will work in the presence of other high-frequency components, as well as other HTHs, and if the HTH can be physically implemented to use part of a RON chain or a pre-existing on-chip ring oscillator without being detected during test.

Future work will also focus on analyzing the effects of the trigger in a larger circuit, in isolation, as will as in the presence of a ring oscillator network (RON). Future work will also focus on developing an improved trigger for the HTH that consumes very little power.

Finally, the analysis in this thesis was performed entirely in simulation. Future work can investigate the amount of data that can be detected when ring oscillators are implemented in physical chips and measured with physical test equipment.

BIBLIOGRAPHY

[1] AnySilicon. (2016) Verification, validation, testing of asic/soc designs – what are the differences? [Online]. Available: https://anysilicon.com/verification-validation-testing-asicsoc-designs-differences/ viii, 4

[2] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE Design Test of Computers*, vol. 27, no. 1, pp. 10–25, Jan 2010. viii, 1, 5

[3] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware trojan attacks: Threat analysis and countermeasures," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1229–1247, Aug 2014. viii, 1, 6

[4] Y. Zhou and D. Feng, "Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing," 2005, zyb@is.iscas.ac.cn 13083 received 27 Oct 2005. [Online]. Available: http://eprint.iacr.org/2005/388 viii, 11, 12

[5] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology — CRYPTO' 99*, M. Wiener, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397. viii, 12, 14, 15, 17

[6] G. E. Moore, "Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff." *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 33–35, Sept 2006. 1

[7] S. Skorobogatov and C. Woods, "Breakthrough silicon scanning discovers backdoor in military chip," in *Proceedings of the 14th International Conference on Cryptographic Hardware and Embedded Systems*, ser. CHES'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 23–40. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33027-8_2 1

[8] A. K. Khan and H. J. Mahanta, "Side channel attacks and their mitigation techniques," in *2014 First International Conference on Automation, Control, Energy and Systems (ACES)*, Feb 2014, pp. 1–4. 1, 4, 11

[9] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug 2014. 1

[10] M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: Models, methods, and metrics," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1283–1295, Aug 2014. 1

[11] J. Dofe, Q. Yu, H. Wang, and E. Salman, "Hardware security threats and potential countermeasures in emerging 3d ics," in *2016 International Great Lakes Symposium on VLSI (GLSVLSI)*, May 2016, pp. 69–74. 1

[12] J. Zhang and Q. Xu, "On hardware trojan design and implementation at register-transfer level," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, June 2013, pp. 107–112. 2

[13] S. Mitra, H. . P. Wong, and S. Wong, "The trojan-proof chip," *IEEE Spectrum*, vol. 52, no. 2, pp. 46–51, February 2015. 4

[14] S. Adee, "The hunt for the kill switch," *IEEE Spectrum*, vol. 45, no. 5, pp. 34–39, May 2008. 4

[15] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting malicious inclusions in secure hardware: Challenges and solutions," in *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, June 2008, pp. 15–19. 5

[16] Y. Alkabani and F. Koushanfar, "Extended abstract: Designer's hardware trojan horse," in *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, June 2008, pp. 82–83. 6

[17] M. Tehranipoor and C. Wang, *Introduction to hardware security and trust*, 2012. 7

[18] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware trojans: Lessons learned after one decade of research," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 22, no. 1, pp. 6:1–6:23, May 2016. [Online]. Available: http://doi.acm.org/10.1145/2906147 8

[19] S.-J. Wang, J.-Y. Wei, S.-H. Huang, and K. S. Li, "Test generation for combinational hardware trojans," in *2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST)*, Dec 2016, pp. 1–6. 9

[20] M. Banga and M. S. Hsiao, "Trusted rtl: Trojan detection methodology in pre-silicon designs," in *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, June 2010, pp. 56–59. 9

[21] X. Zhang and M. Tehranipoor, "Case study: Detecting hardware trojans in third-party digital ip cores," in *2011 IEEE International Symposium on Hardware-Oriented Security and Trust*, June 2011, pp. 67–70. 9

[22] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," in *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, June 2008, pp. 51–57. 10

[23] R. Rad, J. Plusquellic, and M. Tehranipoor, "Sensitivity analysis to hardware trojans using power supply transient signals," in *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, June 2008, pp. 3–7. 10

[24] X. Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic, "Hardware trojan detection and isolation using current integration and localized current analysis," in *2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*, Oct 2008, pp. 87–95. 10

[25] J. Rajendran, O. Sinanoglu, and R. Karri, "Regaining trust in vlsi design: Design-for-trust techniques," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1266–1282, Aug 2014. 10

[26] S. Bhunia, S. Ray, and S. Sur-Kolay, *Fundamentals of IP and SoC Security: Design, Verification, and Debug*, 1st ed. Springer Publishing Company, Incorporated, 2017. 10

[27] J. Rajendran, O. Sinanoglu, and R. Karri, "Is split manufacturing secure?" in *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2013, pp. 1259–1264. 10

[28] S. Mangard, "Hardware countermeasures against dpa ? a statistical analysis of their effectiveness," in *CT-RSA*, 2004. 15

[29] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, 2007. 15, 17

[30] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure dpa resistant asic or fpga implementation," in *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 1, Feb 2004, pp. 246–251 Vol.1. 18

[31] ISCAS. Iscas-85 c432 27-channel interrupt controller. [Online]. Available: http://web.eecs.umich.edu/~jhayes/iscas.restore/c432.html 19, 20, 30, 31

[32] ——. Iscas-85 c499/c1355 32-bit single-error-correcting circuit. [Online]. Available: http://web.eecs.umich.edu/~jhayes/iscas.restore/c499.html 19, 20, 30, 37

[33] Si2. (2008) Nangate freepdk45 generic open cell library. [Online]. Available: https://projects.si2.org/openeda.si2.org/projects/nangatelib 20

[34] Cadence. Virtuoso schematic editor. [Online]. Available: https://www.cadence.com/content/cadence-www/global/en_US/home/tools/custom-ic-analog-rf-design/circuit-design/virtuoso-schematic-editor.html 20

[35] ISCAS. Iscas high-level models. [Online]. Available: http://web.eecs.umich.edu/~jhayes/iscas.restore/benchmark.html 20

[36] Synopsys. All products. [Online]. Available: https://www.synopsys.com/company/resources/products-directory.html 20

[37] X. Xu. Library compiler by synopsys. [Online]. Available: http://www.utdallas.edu/~Xiangyu.Xu/lc/ 20

[38] Synopsys. Dc ultra. [Online]. Available: https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/dc-ultra.html 20

[39] T. Manikas. Rtl compiler - synopsys design compiler. [Online]. Available: https://s2.smu.edu/~manikas/CAD_Tools/SDC/SynopsysDesignCompiler.html 20

[40] Synopsys. Primetime static timing analysis. [Online]. Available: https://www.synopsys.com/implementation-and-signoff/signoff/primetime.html 21

[41] LibreOffice. Libreoffice. [Online]. Available: https://www.libreoffice.org/ 21

[42] MathWorks. Matlab. [Online]. Available: https://www.mathworks.com/products/matlab.html 21

[43] L. Goldstein, "Controllability/observability analysis of digital circuits," *IEEE Transactions on Circuits and Systems*, vol. 26, no. 9, pp. 685–693, September 1979. 23

[44] F. Van Veen, "An introduction to ic testing," *IEEE Spectr.*, vol. 8, no. 12, pp. 28–37, Dec. 1971. [Online]. Available: https://doi.org/10.1109/MSPEC.1971.5217887 24

[45] S. Saha, R. S. Chakraborty, and D. Mukhopadhyay, "Testability based metric for hardware trojan vulnerability assessment," in *2016 Euromicro Conference on Digital System Design (DSD)*, Aug 2016, pp. 503–510.

[46] D. M. Shila and V. Venugopal, "Design, implementation and security analysis of hardware trojan threats in fpga," in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 719–724.

[47] H. Salmani and M. Tehranipoor, "Analyzing circuit vulnerability to hardware trojan insertion at the behavioral level," in *2013 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, Oct 2013, pp. 190–195.

[48] S. R. Priya, P. Swetha, D. Srigayathri, N. Sumedha, and M. Priyatharishini, "Hardware malicious circuit identification using self referencing approach," in *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, Aug 2017, pp. 1–5.

[49] J. Dofe, Q. Yu, H. Wang, and E. Salman, "Hardware security threats and potential countermeasures in emerging 3d ics," in *2016 International Great Lakes Symposium on VLSI (GLSVLSI)*, May 2016, pp. 69–74.

[50] D. Karaklajić, J. Schmidt, and I. Verbauwhede, "Hardware designer's guide to fault attacks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 12, pp. 2295–2306, Dec 2013.

[51] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting malicious inclusions in secure hardware: Challenges and solutions," pp. 15–19, 07 2008.

[52] C. A. Mack, *Fundamental Principles of Optical Lithography*, 2007.

[53] X. Zhang and M. Tehranipoor, "Ron: An on-chip ring oscillator network for hardware trojan detection," in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–6.

[54] S. Docking and M. Sachdev, "An analytical equation for the oscillation frequency of high-frequency ring oscillators," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, pp. 533–537, March 2004.

[55] D. Semiconductor. (2017) High frequency power measurements: Are your oscilloscope and probes telling you the whole truth. [Online]. Available: https://www.mouser.com/pdfDocs/d3semiconductor-hfmeasurementsinpowerswitching-july-2017.pdf

[56] Synopsys, "Custom waveview." [Online]. Available: https://www.synopsys.com/verification/ams-verification/custom-waveview.html 21

[57] ——, "Hspice." [Online]. Available: https://www.synopsys.com/verification/ams-verification/hspice.html 21

[58] Cadence. (2014) Virtuoso analog design environment family. [Online]. Available: https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/tools/custom-ic-analog-rf-design/virtuoso-analog-design-fam-ds.pdf 21

[59] Y. Nasser, J. Prévotet, M. Hélard, and J. Lorandel, "Dynamic power estimation based on switching activity propagation," in *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, Sept 2017, pp. 1–2. 12